uc3m | Universidad Carlos III de Madrid

Grado Universitario en Ingeniería Informática
2017-2018

Trabajo Fin de Grado

PROTECCIÓN DE CREDENCIALES EN TRABAJOS FIN DE GRADO: ANÁLISIS Y ELABORACIÓN DE MATERIALES DIDÁCTICOS

José Luis Parra Olmedilla

Tutor

Ana Isabel González-Tablas Ferreres

Leganés 18/10/2018

Índice

Índice de Tablas

Índice de ilustraciones

Agradecimientos

Me gustaría agradecer a mi familia el apoyo recibido durante mi etapa universitaria. A mis amigos, que me han ayudado en lo que han podido durante este camino. A los nuevos amigos que me ha aportado la universidad, compartiendo tanto los buenos momentos, como los malos en esta época.

Por último, agradecer a los profesores de la universidad por los conocimientos que nos han proporcionado. En especial, a mi tutora Ana Isabel, que me ha guiado y aportado nuevos conocimientos durante el desarrollo de este trabajo.

Resumen

En esta memoria explicamos el desarrollo de un sistema con el principal objetivo de proteger las credenciales de los usuarios. Ha sido diseñado para ser altamente integrable en los diferentes sistemas, actuando más como un módulo que como una aplicación como tal. Para alcanzar este objetivo, se han analizado una serie de protocolos y tipos de autenticación que nos han ayudado a adquirir el conocimiento necesario para el desarrollo del sistema.

Se incluye también una investigación sobre los Trabajos de Fin de Grado (TFG) previamente publicados llevada a cabo gracias a los registros online de la biblioteca de la Universidad. Este estudio nos ayudó a darnos cuenta de cómo es de importante la protección de las credenciales, así como de su común presencia en otros sistemas.

En este punto ya tenemos una idea de la importancia de la protección y algunos de los enfoques de otros desarrolladores, por lo que se comienza el diseño y el análisis del sistema. El diseño consta de un módulo de seguridad implementado en JAVA usando la librería ESAPI, que permite realizar un mantenimiento seguro de los usuarios. Una vez implementado, se definen una serie de pruebas que prueben el cumplimiento de los requisitos.

Finalmente, la memoria incluye la planificación temporal y económica que se ha llevado para la realización de este proyecto, así como un manual de usuario que explica cómo usar el módulo con un enfoque más práctico.

1 Introducción

Hoy en día todas las personas tenemos algún tipo de información personal almacenada en entornos ajenos a nosotros tales como aplicaciones, redes sociales, etc. En el momento actual, la tendencia a tener datos personales distribuidos en internet está aumentando. No solo el peligro está cuando realizamos una conexión a internet, actualmente por ejemplo, los dispositivos móviles que portamos diariamente contienen una gran cantidad de datos personales.

Por lo tanto, se ha convertido en otra manera de recabar información personal, e incluso lograr un beneficio económico con la venta o extorsión de ello. Para evitar estas vulnerabilidades, es necesario que nuestros sistemas tengan un tratamiento de la información personal adecuado. Es decir, utilizar los protocolos y mecanismos adecuados para mantener la información personal segura. Sin embargo, siguen existiendo sistemas que no realizan un tratamiento de los datos personales aunque actualmente existen múltiples herramientas que facilitan la implementación de autenticación en los sistemas.

Los datos personales mencionados anteriormente, corresponden generalmente a las credenciales de usuario. Las credenciales de usuario son datos que permiten identificar a los usuarios en algún sistema. Cómo ejemplo, la manera más común suele ser introducir un usuario y contraseña. Es decir, estos serán los datos que debemos proteger para garantizar la información de los usuarios.

1.1 Motivación

La experiencia previa de otros años, es que una gran parte de los TFG maneja credenciales de usuario, pero en muchos casos, esta información no es tratada como se debe. Se almacenan datos en claro o no se define una posible manera de almacenar estos datos.

La aplicación de seguridad en la información personal también se debe aplicar en los TFG. Es importante que al diseñar e implementar cualquier sistema, por pequeño que sea, como por ejemplo es el caso de los TFGs, debemos estar mentalizados que pueden contener información personal y que un mal tratamiento de estos datos puede tener graves consecuencias tanto para nosotros, como para los futuros usuarios del sistema.

Contar con cierta experiencia a la hora de proteger las credenciales de usuario prepara mejor a los estudiantes, permitirá que los estudiantes sean conscientes de esta necesidad y su criticidad, y les preparará mejor para afrontar su carrera laboral, beneficiando en última instancia a la sociedad. Sin embargo, muchos de los TFGs no tienen por primer objetivo asegurar el sistema que se presenta.

Por lo tanto, sería interesante poder proporcionar un pequeño módulo de autenticación, ya que probablemente en la mayoría de casos, la falta de tiempo a la hora de realizar los trabajos, puede ser responsable de que no se implementen dichos mecanismos de seguridad.

1.2 Objetivo

En este documento, realizaremos un estudio de las actuales posibilidades que existen a la hora de proteger las credenciales, intentando cubrir la mayor gama de tipos. Posteriormente se realizará un estudio con TFGs de la universidad para obtener

información de la protección que se realiza en estos. Una vez tenemos todos los datos anteriores, se procederá a implementar una solución que pueda facilitar el trabajo en el futuro a otros compañeros. Además se crearan materiales didácticos para facilitar el uso de estas soluciones.

La finalidad principal de este trabajo es por tanto proporcionar una solución para proteger las credenciales de usuarios en los TFGs, en concreto utilizando el lenguaje Java. Para ello, es necesario analizar el estado actual y las posibles soluciones que pueden implementarse, y adecuar esta solución a los TFG. Es decir, una solución estándar que permita aplicarse en cualquiera de los trabajos a pesar de su tamaño y requisitos.

Por lo tanto, el proceso a seguir será el siguiente:

1. Análisis del estado actual de la protección de credenciales.
2. Estadísticas sobre la protección de credenciales en TFGs.
3. Estudio de las posibles soluciones.
4. Implementar una solución estándar para permitir su uso en la mayoría de TFGs en el lenguaje de programación Java.
5. Creación de materiales didácticos.
6. Fomentar el uso de herramientas que permitan una protección de credenciales.

1.3 Estructura del documento

El documento comienza con una breve introducción en la que nos encontramos donde mostramos una pequeña descripción del trabajo que vamos a realizar. Además estará acompañado por los objetivos que tratamos de solucionar con este trabajo y la motivación de la realización de este trabajo.

Tras la introducción, continuamos con una investigación del estado del arte. Esto, nos permitirá ver distintas herramientas y cómo ha evolucionado la protección de credenciales a lo largo del tiempo. Es una parte importante del trabajo, ya que el conocimiento de la situación de herramientas, métodos de autenticación o protocolos, nos permite encontrar una solución más óptima a nuestro problema.

Una vez finalizado el estado del arte, se exponen las estadísticas extraídas de TFGs de años anteriores. Estas estadísticas, nos proporcionan una idea del impacto que puede tener el sistema que vamos a proporcionar y si realmente puede ser un buen proyecto a desarrollar.

A continuación analizaremos las necesidades que tendrá nuestro sistema. Para ello se tomarán requisitos de usuario que serán, posteriormente convertidos en requisitos software que cubrirán la funcionalidad de nuestro sistema. Además se tendrá en cuenta las características de los entornos en los que se va a trabajar.

Tras analizar las necesidades del sistema, es necesario dar forma a este mediante el diseño. En él se establecerá la arquitectura, se diseñaran los casos de uso y se dará forma a una interfaz sencilla que permita un funcionamiento fácil del sistema.

Después se describirá de una manera breve la implementación que se lleva a cabo, en las que se explica cómo se ha desarrollado el sistema. Además se incluye como implantar el sistema en otros sistemas.

Para validar el software, se realizará una evaluación, en la que se incluye un plan de pruebas que comprueba el correcto funcionamiento del sistema. Tras esto, se desarrolla lo referente a la gestión del proyecto en las que se incluyen metodología, presupuesto, planificación y marco legal.

Seguidamente, finalizamos con las conclusiones en las que se plasmara la experiencia de haber desarrollado el trabajo y posibles trabajos futuros que se pueden desarrollar a partir de este trabajo.

Finalmente, para facilitar el uso del sistema a los usuarios, en un anexo se realiza un material didáctico que está compuesto por un manual de usuario. El manual estará compuesto por capturas de pantallas y pequeños párrafos para que pueda ir siguiendo la guía facilitando el uso.

2 Estado del arte

Las credenciales de usuario son datos que permiten identificar a los usuarios. Las credenciales de usuario, se utilizan para la autenticación en los sistemas permitiendo así, el acceso a los usuarios de manera segura. La autenticación, la noción de que eres quien dices ser, es requerida en los puntos de entrada a las aplicaciones que realizan entregas o transacciones de datos. En dependencia de la gravedad del daño que puede ocurrir si un atacante realiza una transacción maliciosa utilizando la identidad de usted, es que se requieren métodos fuertes de autenticación. Esto implica que todos los sistemas, utilicen mecanismos para evitar el robo de estos datos y permitir a los usuarios tener la confianza de que el sistema que utilizan es seguro.

La protección de credenciales siempre ha sido un problema para los programadores, pero hay una clara evolución en la identificación de usuarios. Desde el método más antiguo como pueden ser los usuarios y contraseñas, hasta los métodos más modernos, como puede ser la utilización de biométricos. A continuación, mostramos una lista con una pequeña descripción:

- Identificación y contraseña: Es el método más básico a la hora de autenticar un usuario. Por ejemplo, es el método de acceder a aula global.
- Intercambio de claves: Es un método más complejo, se utilizan pares de claves pública y privada, para comprobar la autenticación de las entidades.
- OTP: Se utilizan "contraseñas de un único uso", es decir, la contraseña se establece para un periodo de tiempo y solo permite una única sesión.
- Utilización de biométricos: Permite la autenticación a partir de características físicas del usuario.
    - Huella dactilar: Se realiza la autenticación mediante una o varias huellas de los dedos.
    - Iris: Se capta el iris para lograr la autenticación.
    - Reconocimiento facial: Capta la forma y rasgos de la cara para identificar a la persona que intenta autenticarse.
- RFID activo: Permite la autenticación a distancia ya que posee una alimentación propia que le permite dialogar con el elemento que trata de autenticar.

Sin embargo, para el problema que tratamos de solucionar en este trabajo, nos centraremos únicamente en los sistemas de autenticación PAP o lo que es lo mismo, autenticación con usuario y contraseña. Para plantear una solución, tenemos que tener en cuenta las vulnerabilidades habituales de este tipo de autenticación:

- Contraseñas sencillas: Permiten de manera sencilla, aunque las credenciales de usuario estén protegidas, obtener la contraseña.
- Contraseñas por defecto: En muchas ocasiones los sistemas otorgan de manera automática una contraseña al usuario, que a pesar de indicar que está debe cambiarse, en algunos casos el usuario hace caso omiso a estas indicaciones. En algunos casos, se imponen unas contraseñas por defecto para que los

administradores puedan acceder de manera remota en caso de tener algún problema.

- Contraseñas compartidas: En sistemas en los que varios usuarios tienen que utilizar el mismo dispositivo y no soportan varias cuentas de usuario, obligan a los usuarios a utilizar la misma contraseña. Esto genera varios problemas, el primero es que en ningún momento el usuario real esta autenticado, si no que todo está asociado a un usuario ficticio del sistema, por lo que nunca se podrá ver la autoridad real de los hechos realizados por el usuario. Por otro lado, aunque un usuario deje de utilizar el sistema y no tenga autorización a utilizarlo, va a seguir teniendo el conocimiento del usuario y podría volver a aparecer.
- Uso de un mayor número de aplicaciones con credenciales: Cada vez es mayor el número de aplicaciones y usuarios que deben disponer de credenciales para acceder a ciertas aplicaciones. Esto implica un mayor traspaso de información de autenticación por la red, y en algunos casos se suele utilizar la misma contraseña en las aplicaciones para facilitar el recuerdo, que en caso de no estar correctamente asegurada puede suponer el descubrimiento de las credenciales.
- Cuentas con privilegios: En todos los sistemas se incluyen cuentas de administrador que permiten un acceso total a los sistemas. Estas cuentas suelen ser administradas por administradores de la aplicación, pero un mal uso de este tipo de cuenta o un usuario que consiga utilizarla, podría obtener credenciales en caso de no estar protegidas de manera correcta. [1]

Para que nuestros sistemas sean seguros y protejan de manera correcta la información, se debe minimizar al máximo las vulnerabilidades mencionadas anteriormente. Por lo general, el esfuerzo que se utiliza en seguridad en algunos sistemas es inferior al que se debería, ya que es un proceso costoso a nivel económico. Sin embargo, existen determinadas regulaciones y normativas que regulan la seguridad de estos sistemas. Por lo tanto, los sistemas deberían cumplir al menos estas regulaciones que permiten reconocer el sistema cómo una autoridad segura.

Además del esfuerzo económico mencionado anteriormente, la implementación de medidas de seguridad, puede complicar la gestión del sistema, su desarrollo y aumentar el grado de dificultad que puede tener realizar cambios en la aplicación.

En un nivel más complejo, con el paso del tiempo se ha buscado métodos de autenticación más seguros, en las que necesites más elementos además de tus credenciales para identificar de manera correcta al usuario. Un ejemplo es la autenticación multifactor (MFA) qué es un sistema de seguridad que requiere más de una forma de autenticación para verificar la legitimidad de una transacción.

La ventaja que tiene los métodos de autenticación MFA consiste en el uso de credenciales independientes. Es decir, aunque se descubra por ejemplo el usuario y la contraseña de un usuario, el usuario que está suplantando, no va a poder acceder ya que además de esta credencial, necesita por ejemplo algún tipo de información biométrica como puede ser una huella dactilar, o por ejemplo el usuario tiene que tener una cierta ubicación. A pesar de no ser un método de autenticación que trate las credenciales de una manera más segura, permite proteger la autoridad del usuario necesitando cumplir un mayor número de condiciones.

A pesar de que las empresas o desarrolladores tienen el conocimiento total que los sistemas tienen que tratar la información sensible del usuario, se hace caso omiso a ello hasta que aparece algún tipo de problema. En la actualidad, los ataques a sistemas han aumentado y en muchos casos se producen debido a la poca protección que tienen los sistemas. Sin embargo, en la actualidad, está habiendo un auge de las identity and access management (IAM), o control de acceso e identidad.

Los IAM son frameworks que facilitan el manejo de las identidades de usuario. Es decir, facilitan la labor de securizar los accesos de los usuarios proporcionando un módulo que protege las credenciales e identidades de estos. Estos frameworks, permiten acceso a los usuarios basándose en un acceso por roles otorgados por los administradores.

Los sistemas utilizados para la gestión de identidad y acceso incluyen sistemas *single-sign-on*, autenticación multifactor y gestión de acceso. Estas tecnologías también ofrecen la capacidad de almacenar de forma segura los datos de identidad y perfil, así como las funciones de gobierno de datos para garantizar que solo se compartan los datos que son necesarios y relevantes.

Estos sistemas, tienen que tener la capacidad de capturar y registrar el inicio de sesión de los usuarios permitiendo así su supervisión. Además, estos sistemas deben simplificar el proceso de aprovisionamiento y configuración de la cuenta del usuario, disminuyendo así posibles errores de seguridad al configurar los usuarios. Al igual que se debe tener en cuenta la facilidad en la creación, deben proveer una correcta manera de edición. [2]

En un futuro cercano, con el crecimiento del internet de la cosas (IOT), un número mayor de dispositivos y sistemas estarán conectados a las redes. Todos estos sistemas, tendrán información personal, un objetivo que puede ser muy jugoso para los atacantes. Por lo tanto, en un futuro cercano, aparecerán nuevos módulos de seguridad para este tipo de elementos. Otro problema en un futuro cercano es el almacenamiento de datos en la nube. Un ejemplo en la actualidad puede ser los sistemas que utilizan protocolos como O-Auth 2.0 (Open Authorization), el cual es un protocolo que permite flujos simples de autorización para sitios web o aplicaciones informáticas.

2.1 Estudio de posibles soluciones

En este apartado describiremos de manera detallada las posibles soluciones que podemos tener al problema que queremos resolver. Como comentamos en el apartado anterior, la solución que queremos encontrar sería para aplicaciones con credenciales de usuario de usuario y contraseña. Por lo tanto la autenticación con biométricos no será importante en este apartado.

A continuación, describimos una serie de protocolos de autenticación básicos que permiten autenticarse de manera sencilla. Para ponernos en contexto, la autenticación es un proceso criptográfico que tiene el propósito de autentificar entidades que desean comunicarse de forma segura, es decir, identificar correctamente a los usuarios que van a utilizar el sistema [3]. Algunos protocolos de autenticación son:

2.1.1 PAP (Protocolo de autenticación de contraseña)

Este protocolo es el más simple de los existentes. Para autenticar al usuario, el sistema envía el usuario y la contraseña sin cifrar, es decir, sin ningún cifrado que proteja estos datos. Por lo tanto, en la actualidad, no es recomendable utilizar PAP ya que es fácil interceptar el intercambio de la contraseña entre el sistema y el usuario. [3]

2.1.2 CHAP (Protocolo de autenticación por desafío mutuo)

El método CHAP está basado en el protocolo PAP pero se introduce seguridad en el intercambio de credenciales. Con CHAP, el sistema envía un desafío al cliente y el cliente mediante una función hash utilizando MessageDigest-5 (MD5) calcula el resumen del desafío y un hash con la contraseña del usuario que será enviado al sistema para verificar su acceso. Antes de continuar, aclarar que el algoritmo MD5 se ha roto, y por lo tanto, no es válido para la autenticación.

El sistema, al tener acceso a los datos de usuario entre ellos la contraseña, puede calcular el hash de está para verificar si es correcto el usuario. Si el resultado del desafío que envía el cliente y la contraseña corresponde con los calculados por el sistema, el sistema permitirá acceder al usuario. [3]

2.1.3 SPAP (Protocolo de autenticación de contraseña de shiva)

El protocolo SPAP es un protocolo similar al protocolo de CHAP, sin embargo es más seguro que este. El usuario envía la contraseña al sistema cifrada, y el sistema en este protocolo, descifra esta contraseña y la compara con la que tiene almacenada en claro. Esto es posible ya que SPAP utiliza un protocolo en cifrado bidireccional, que permite descifrar la contraseña. [3]

2.1.4 MS-CHAP y MS-CHAP v2

MS-CHAP es un mecanismo de autenticación que desarrollo Microsoft que permite la autenticación entre distintas estaciones de trabajo conectadas mediante una red LAN. Este mecanismo se diferencia de los mencionados anteriormente ya que para la autenticación no necesita contraseña, solo utiliza desafíos e identificadores de sesión. [3]

El sistema envía un reto que contiene el identificador de sesión y una cadena de valores aleatorios. El usuario envía al servidor el usuario que le corresponde acompañado de una función resumen del desafío que le proporciona el sistema. Cuando el sistema recibe esta información verifica si esta es correcta, y si es así, el usuario quedará autenticado. [4]

2.1.5 Protocolo de autenticación extensible (EAP)

El protocolo EAP permite validar conexiones de acceso remoto. Para ello, el usuario que intenta acceder a un punto de acceso, realiza una solicitud al servidor (en la mayoría de los casos, es un servidor RADIUS), que debe ser compatible con el tipo de mecanismo EAP con el que se quiere autenticar.

Como mencionamos anteriormente, el protocolo de autenticación EAP tiene distintos mecanismos para la autenticación. Pero como hemos indicado antes, el servidor que autentique la conexión, tiene que tener el mismo mecanismo que el cliente que intenta autenticarse. [3] [5]

2.1.5.1 EAP-TLS

Este tipo de mecanismo utiliza la protección que actualmente ofrece la capa de transporte en las redes de comunicaciones. Este mecanismo de seguridad es Transport Layer Security (TLS). Al igual que TLS este mecanismo utiliza certificados para la autenticación en las conexiones.

Gracias a la codificación de datos que permite realizar TLS, puede autenticar la comunicación que se realice entre el usuario y el servidor. Además, como puede verificar la autenticación mutua entre ellos, posibilita la negociación de un algoritmo y claves para establecer la comunicación.

2.1.5.2 EAP-RADIUS

Como hemos visto en apartados anteriores, RADIUS será el servicio que autenticara la comunicación ente el cliente y el servidor. Los servicios que proporcionan se conocen cómo "AAA" (autorización, autenticación y auditoría). Se utiliza cuando un cliente intente autenticarse en el sistema. Por lo general, los servidores RADIUS son los más utilizados en este protocolo. A continuación describimos los distintos servicios que proporciona RADIUS: [5]

- Autenticación: Comprueba que el usuario y la contraseña se encuentren informados en la base de datos local. Si estos datos son encontrados se procederá a la autorización.
- Autorización: Comprueba si tiene permisos para acceder al recurso. Se asigna una dirección IP al cliente de acceso telefónico.
- Auditoría: Cómo describe el servicio, se extrae la información referente a la conexión establecida.

2.1.6 Kerberos

La motivación para desarrollar Kerberos es la de realizar un protocolo que se puede utilizar en redes inseguras. Es decir, es un protocolo que a pesar de dos sistemas se encuentren en una red insegura, pueden autenticarse de manera segura.

Este protocolo requiere otros servicios de confianza para poder conseguir su objetivo, ya que Kerberos, permite la autenticación mutua identificando correctamente tanto al cliente, cómo al servidor. Además Kerberos protege contra ataques de reenvío y protege de la interceptación de mensajes. [3]

2.1.7 Single Sign On (SSO)

Los métodos explicados anteriormente son de finales del siglo XX y primeros años del siglo XXI, por lo que pese a que algunos se utilizan actualmente aún, han pasado a formar parte de métodos de autenticación más complejos. Por lo tanto, tener un conocimiento de ellos es necesario para poder crear o afrontar métodos más complejos.

Con el mayor uso de sistemas en los que hay que utilizar contraseñas, el usuario tiene qué almacenar un mayor número de contraseñas, o lo que podría ser un riesgo, utilizar la misma contraseña para todos los sistemas. Como vimos anteriormente, puede permitir si se detecta la contraseña en una de los sistemas, obtener la del resto de sistemas.

Este problema se puede mejorar mediante el uso de protocolos *Single-Sign-On* (SSO). El SSO permite acceder a diferentes servicios con una única identidad, lo que permite mejorar la gestión de los usuarios en los sistemas.

Con el paso del tiempo, los usuarios empiezan a utilizar un mayor número de sistemas que requieren la utilización de credenciales de usuario. Esto puede llegar a ser un problema, ya que se deben recordar una gran cantidad de contraseñas. Sin embargo, el *Single-Sign-On* permite acceder a distintos sistemas utilizando únicamente una identidad. A continuación explicamos en profundidad que es el SSO, sus características y tipos.

Por lo tanto, los SSO mediante una única autenticación, con la identidad almacenada en el SSO, pueden acceder a diferentes sistemas o herramientas. Es decir, en lugar de tener que estar iniciando sesión en distintos sistemas, la identidad única consigue que este la sesión iniciada en todas las herramientas que soporten el SSO y por lo tanto, no deberán introducir de nuevo sus credenciales. [6]

Por ejemplo, las cuentas de Gmail, son un ejemplo de este tipo de autenticación. Cuando iniciamos sesión con nuestra cuenta en el explorador, podemos acceder a todas las aplicaciones de Google, como puede ser Drive, Maps…

2.1.7.1 Características de Single Sign On

La autenticación mediante SSO es un tipo de autenticación muy distinto a los vistos hasta ahora. Este tipo de autenticación está orientado a sistemas con credenciales de usuario. Sus principales características son las siguientes:

- Como hemos mencionado, permite el acceso a múltiples sistemas con el uso del mismo usuario y contraseña. Por lo tanto, se tiene que realizar una sincronización periódica de estos usuarios.
- Los identificadores que utilizan SSO mejoran la seguridad de la red debido a que estas, son capaces de identificar al usuario de manera inequívoca. Esto provoca que cumpla normas más estrictas de seguridad. Además la información sobre los usuarios que proporciona el SSO, se envía cifrada.
- Disminuye el número de credenciales que deben recordar los usuarios, ya que con una única identidad, puede acceder a distintos sistemas.
- El usuario al estar identificado en uno de los servicios del SSO, no deberá iniciar sesión al utilizar un servicio distinto. [6]

2.1.7.2 Tipos de SSO

- Enterprise *Single-Sign-On* (E-SSO): Este SSO recoge los datos del inicio de sesión de un servicio primario, y es usado posteriormente para el resto de servicios que se pueden utilizar dentro de la estructura del SSO.
- Web *single-sign-on* (Web-SSO): Es similar al tipo anterior adaptado a páginas web, pero en este caso se extrae la información del inicio de sesión de la primera página que se visita y estos datos son utilizados para el acceso al resto de páginas.

Para extraer la información de los inicios de sesión mencionados anteriormente, se hace uso de un proxy que recoge los datos. En caso de perder los datos de la autenticación, pedirá de nuevo al usuario que se autentique.

- Identidad federada: Identifica a los usuarios gracias a un conjunto de estándares. El objetivo del uso de estos estándares es permitir el uso de estos usuarios en distintos sistemas que cumplen el estándar.
- Open ID: Es un protocolo que puede implementar los SSO e identifica a los usuarios a través de XRI. En el siguiente apartado lo veremos en profundidad.

Una vez hemos definido que son los SSO, características y tipos, encontramos los conceptos de identidad federada y Open ID. Vamos a ver de una manera más concreta que son estos conceptos y cómo están relacionados entre sí. [6]

La identidad federada significa vincular y usar las identidades electrónicas que tiene un usuario en varios sistemas de administración de identidades.

Es decir, las credenciales de usuarios pasan a estar en un entorno externo a la propia aplicación, sino que está, extraerá de los datos de un sistema de gestión de identidades que tendrá almacenada la identidad electrónica del usuario. Este sistema de gestión de identidades, debe ser confiable y la aplicación aceptara estos datos para autenticar.

Un ejemplo de este tipo de identidades federadas son las innumerables aplicaciones que permiten el inicio de sesión mediante alguna de las redes sociales. La más habitual suele ser la autenticación mediante Facebook, ya que es una de las redes sociales más extendidas en la actualidad y es la que más usuarios tiene.

Este enfoque permite el desacoplamiento de las funciones de autenticación y autorización. También hace que sea más fácil centralizar estas dos funciones en la empresa para evitar una situación en la que cada aplicación tenga que administrar un conjunto de credenciales para cada usuario. También es muy conveniente para los usuarios, ya que no tienen que mantener un conjunto de nombres de usuario y contraseñas para cada aplicación que utilizan.

Para implementar este tipo de soluciones se utilizan una serie de protocolos. Uno de ellos es Open ID que veíamos en los tipos de SSO, pero además existen otros protocolos cómo pueden ser OAuth o SAML.

2.1.8 OpenID

OpenID cambia la manera de identificarnos, ya no es necesario un usuario. Podremos ser identificados en servidores que soporten este estándar únicamente a través de una URL o en la actualidad con un XRI.

Esto conlleva un cambio de mentalidad en los usuarios, ya no hay que crear cuentas de usuarios, en su lugar hay que crear identificadores para los servicios que soporten OpenID. Para ellos se deberá registrar en proveedores de identidad.

Sin embargo, el estándar OpenID no establece un mecanismo de autenticación, lo que puede suponer un problema para ciertas comunicaciones e intercambios de información. Por lo tanto, el mecanismo de seguridad dependerá de la confianza del proveedor de identidad en el que se encuentre el cliente. [7]

2.1.8.1 Identificadores de OpenID

Como hemos podido ver, la identificación se realiza a través de URL o XRI. Mediante URL te podrás autenticar usando una URL propia, modificando el código

HTML para que permita la identificación mediante OpenID. Otra manera de identificarte a través de URL es registrarla en un proveedor de identidades, que sería el equivalente a crear un usuario y contraseña nueva.

Los XRI son identificadores utilizados por la nueva versión de OpenId, que permite identificar identidades digitales de dominio cruzado. Existen dos tipos de XRI de i-nombres e i-números: [7]

- i-nombres: Son equivalentes a los nombres de dominio, por lo tanto, pueden ser reasignables.
- i-numeros: Son valores únicos, es decir, no pueden ser repetidos. Se utilizan para identificar en los servicios de OpenID, por lo tanto, cada i-nombre está relacionado con un i-numero. Por lo tanto, la manera de resolver los i-nombres sería similar a un DNS.

2.1.8.2 Ventajas y desventajas de OpenID

Sin embargo, OpenID puede tener sus puntos negativos. El hecho de centralizar todo en una sola identidad puede ser un problema. Por ejemplo, a partir de la dirección IP, un usuario puede controlar todas las acciones que realiza otro.

Otro problema más básico es la pérdida o robo de contraseña, ya que con esta información, un atacante podría autenticarse en todos los sistemas que pueda autenticarse mediante OpenID. Por lo tanto, se puede convertir en un blanco de ataques.

Por otro lado, tener las cuentas asociadas a un servidor de OpenID puede ser un problema si este servidor cae, ya que se puede dejar de tener acceso a los sistemas.

Como ventaja, como hemos visto anteriormente, facilita el acceso entre distintos sistemas, lo que es cómodo para un usuario. Es decir, recuerda una única cuenta y le permite la autenticación de manera automática en otros sistemas. Aunque el uso de estos protocolos, utilizan es un problema ya que tienen una única identidad, si está identidad está en un servidor correctamente protegido, puede llegar a ser mejor que tener varias contraseñas.

2.1.9 SAML

SAML es un estándar que permite la identificación de los usuarios en varios sistemas únicamente estando registrado en una red que conecta estos sistemas. Por lo tanto, la finalidad es que varios sistemas puedan compartir estas credenciales y el usuario no tenga que estar registrándose en distintos sistemas, es decir, este protocolo permitirá autenticarse a los usuarios en distintos sistemas estando registrado únicamente en un lugar dentro de la red.

Existen distintas versiones de SAML, la actual es la versión 2.0 que lleva vigente desde el año 2005. SAML está basado en lenguaje XML aunque también está formado por perfiles del protocolo y algunos de los mensajes intercambiados. Como mencionábamos anteriormente, es una manera de implementar los SSO.

Al igual que OpenID, únicamente se podrá utilizar este protocolo en sistemas que lo proporcionen. Además, tienen sus propios proveedores de identidad. [8]

2.1.9.1 Identidad en SAML

La identidad en el protocolo SAML, se basa en un documento XML en el que se encuentra la información que permite la identificación de usuario. Este documento, se transmite entre el sistema y el proveedor de identidades permitiendo así que el proveedor lo pueda identificar.

Una vez el proveedor recibe el XML podrá autenticar y autorizar al usuario si confirma que la identidad es confiable. Para ello, el proveedor de identidades envía al sistema que está intentando autenticar una confirmación SAML. Las confirmaciones SAML, utilizaran protocolos de seguridad tal cómo el cifrado para mejorar la seguridad de la autenticación. [8]

2.1.9.2 Ventajas y desventajas

A continuación, mostramos las ventajas del protocolo SAML. [9]

- Es un servicio común para todos los usuarios que utilicen en protocolo, es decir, por ejemplo, tendrán los mismos protocolos de seguridad.
- La información del usuario no debe ser sincronizada constantemente, ni debe ser mantenida por los sistemas.
- Permiten con una única identificación acceder a distintos servicios que utilizan el mismo protocolo, siempre que los usuarios tengan autorización. Por lo tanto, esta manera de acceder a distintos servicios, puede mejorar la confianza de los usuarios.
- Reduce el coste de los sistemas relacionados con el manejo y mantenimiento de usuarios, ya que toda esta información pasa a estar en proveedores de identidad.
- El problema de la gestión de identidades de los sistemas pasa a ser responsabilidad de los proveedores de identidad.

Sin embargo, estos sistemas tienen estas desventajas.

- SAML no está pensado para ser utilizado en dispositivos móviles.
- SAML requiere certificados SSL.

2.1.10 Oauth

OAuth es un protocolo que permite el acceso a usuarios a un sistema sin que este sistema obtenga las credenciales de usuario. Esto es posible gracias a que otra entidad autentica el usuario.

OAuth empezó a ser desarrollado para suplir a OpenID en la aplicación Twitter. Desde entonces tanto está compañía, Google y otras importantes apoyan este protocolo de autenticación. La primera versión de OAuth fue lanzada en el año 2010 y a los dos años se publicó la siguiente gran versión Oauth 2.0. Estas versiones no son compatibles, por lo tanto la mayoría de los sistemas ya implementan la segunda versión de OAuth. Algunos de los servicios que adoptan este tipo de autenticación son Facebook, Github, Linkedin, Azure entre otros.

El ejemplo más común de este tipo de protocolos, son las aplicaciones que permiten la autenticación mediante Facebook. El ejemplo más extendido en los últimos

años es la aplicación Candy Crush Saga. En esta aplicación, se permitía la autenticación mediante un usuario normal y a través de Facebook.

2.1.10.1 Funcionamiento

OAuth 2.0 es un protocolo que permite el acceso a sistemas en los que no ha introducido sus credenciales. Es decir, gracias a este protocolo la autenticación en lugar de ser en el propio sistema, un tercer sistema realiza la autenticación (El caso más común como mencionamos antes es el inicio de sesión con Facebook). El protocolo es desarrollado por el IETF OAuth WG. El funcionamiento de OAuth 2.0 define como un usuario puede compartir información de un sistema A (proveedor de servicio) con un sistema B (consumidor) sin que el sistema B tenga que tener acceso a la identidad o la contraseña del usuario en el sistema A. El diagrama de funcionamiento es el siguiente:

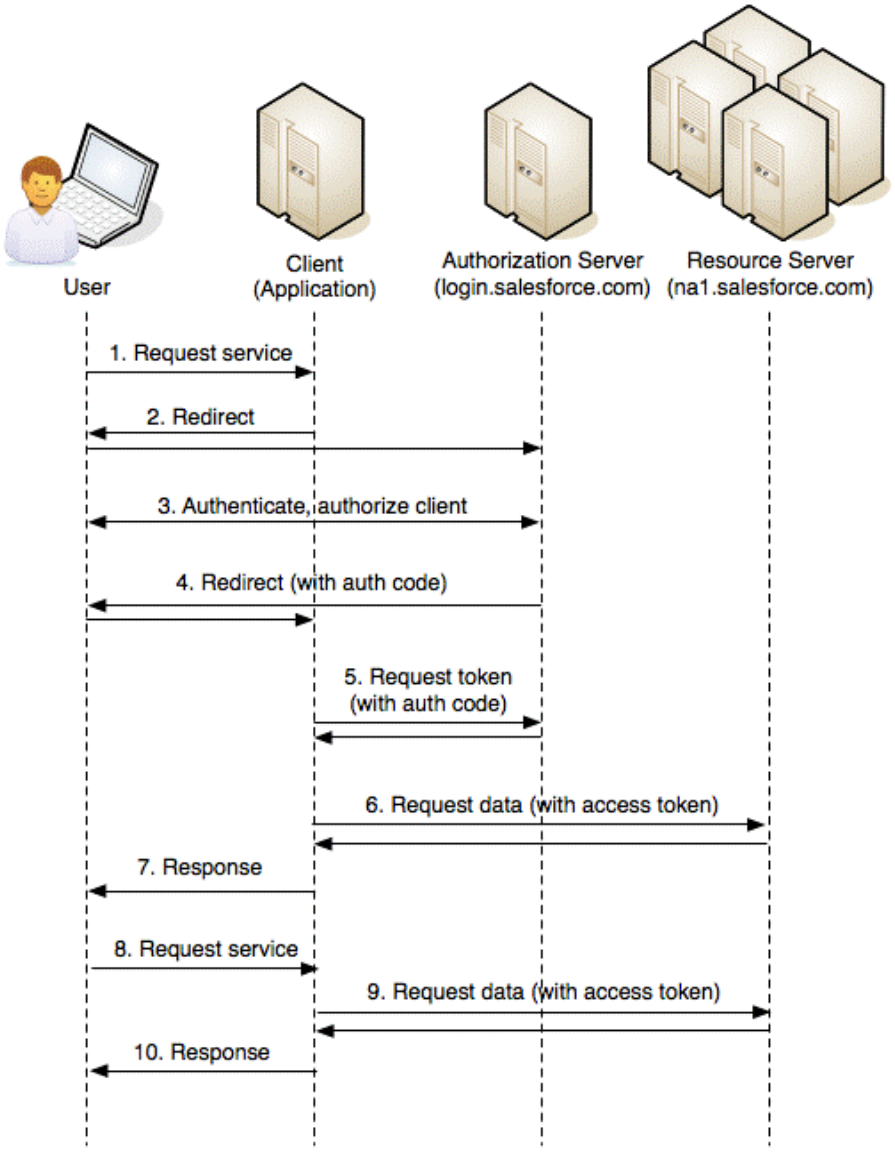

Ilustración 1: Secuencia de básica de uso de OAuth 2.0 [10]

OAuth 2.0 utiliza códigos de acceso temporales que permite que los sistemas que utilizan este tipo de autenticación durante un tiempo limitado. Estos códigos son:

- Código de acceso (*Access Token*).
- Código de autorización (*Authorization Token*).
- Código de refresco (*Refresh Token*).

El uso de un Token de acceso provoca que el robo de este permita el acceso a la aplicación de un usuario que no se autenticado, es decir, un posible atacante. Sin embargo, el protocolo OAuth programado teniendo en cuenta buenas prácticas, se convierte en un protocolo seguro. [10]

2.1.10.2 Ventajas y desventajas

Uno de los problemas de OAuth 2.0 es que este protocolo se basa en que los desarrolladores utilicen TLS, y por lo tanto, no implementan mecanismos de cifrado o firma. Esto implica, que si no hay una protección TLS este protocolo pasa a ser muy vulnerable. Por lo tanto, una de las buenas prácticas que mencionamos en el apartado anterior es que los propios desarrolladores utilicen TLS para securizar este tipo de autenticaciones, ya que TLS proporciona las protecciones que OAuth 2.0 no ha tenido en cuenta.

Otro de los problemas que puede tener OAuth 2.0 es el robo de credenciales a través de un tercer sistema. Es decir, un sistema puede indicar al sistema que autoriza la autenticación del usuario, que necesita ciertos datos del usuario, por ejemplo, para realizar una transferencia. Por lo tanto, este tercer sistema, obtendrá credenciales del usuario que no debería obtener. [11]

2.1.11 Identity and Access Management (IAM)

La administración de identidades y accesos (IAM) en consiste en definir y administrar las funciones y los privilegios de acceso a los usuarios otorgándoles o negándoles dichos privilegios. Esos usuarios podrían ser clientes (gestión de identidad del cliente) o empleados (gestión de identidad del empleado.) El objetivo central de los sistemas IAM es una identidad digital por individuo. Una vez establecida esa identidad digital, debe mantenerse, modificarse y supervisarse a lo largo de cada usuario.

Los sistemas de gestión de acceso a la identidad deben incluir todos los controles y herramientas necesarios para capturar y registrar la información de inicio de sesión de los usuarios, gestionar la base de datos de las identidades de los usuarios y gestionar la asignación y eliminación de los privilegios de acceso. Eso significa que los sistemas utilizados para IAM deben proporcionar un servicio de directorio centralizado con supervisión, así como visibilidad de todos los aspectos de la base de usuarios de la compañía.

Las tecnologías para el acceso y la administración de identidades deberían simplificar el proceso de aprovisionamiento y configuración de la cuenta del usuario. Estos sistemas deberían reducir el tiempo que lleva completar estos procesos a través de un flujo de trabajo controlado que disminuye los errores, al tiempo que permite el cumplimiento automatizado de la cuenta. Un sistema de gestión de identidad y acceso también debería permitir a los administradores ver y cambiar los derechos de acceso al instante.

Estos sistemas también necesitan equilibrar la velocidad y la automatización de sus procesos con el control que los administradores necesitan para monitorizar y modificar los derechos de acceso. En consecuencia, para gestionar las solicitudes de acceso, el directorio central necesita un sistema de derechos de acceso de los usuarios.

Los sistemas IAM deben usarse para proporcionar flexibilidad para establecer grupos con privilegios específicos para funciones específicas, de modo que los derechos de acceso basados en las funciones del trabajo puedan asignarse uniformemente. El sistema también debe proporcionar procesos de solicitud y aprobación para modificar los privilegios, ya que los usuarios con el mismo título y ubicación pueden necesitar acceso personalizado o ligeramente diferente [2] [12]

Cómo resumen, los sistemas IAM brindan a los administradores las herramientas y tecnologías para cambiar la función de un usuario, realizar un seguimiento de las actividades de los usuarios, crear informes sobre esas actividades y aplicar las políticas de forma continua. Estos sistemas están diseñados para proporcionar un medio de administrar el acceso de los usuarios en toda una empresa y para garantizar el cumplimiento de las políticas corporativas y las regulaciones gubernamentales.

2.1.11.1 Ventajas de los IAM

Como mencionamos en el apartado anterior, las tecnologías de IAM se pueden utilizar para iniciar, capturar, registrar y gestionar identidades de usuario y sus permisos de acceso relacionados de forma automática. Esto garantiza que los privilegios de acceso se otorguen de acuerdo con una interpretación de la política y todos los individuos y servicios estén debidamente autenticados, autorizados y auditados.

Debido a que las empresas que administran las identidades de forma adecuada tienen un mayor control del acceso de los usuarios, pueden reducir los riesgos de las violaciones de datos internas y externas.

La automatización de los sistemas IAM permite a los administradores de sistemas operar de manera más eficiente al reducir el esfuerzo, el tiempo y el dinero necesarios para administrar el acceso a sus redes de forma manual o mediante controles de acceso individuales que no están conectados a sistemas de administración centralizados.

El uso de una plataforma común para la gestión de identidad y acceso permite aplicar las mismas políticas de seguridad en todos los diferentes dispositivos y plataformas operativas utilizadas por la empresa. En términos de seguridad, el uso de un marco IAM puede hacer que sea más fácil hacer cumplir las políticas en torno a la autenticación de usuarios, la validación y los privilegios, y abordar problemas relacionados con el arrastre de privilegios.

Al implementar herramientas de administración de acceso de identidad y seguir las mejores prácticas relacionadas, una empresa puede obtener una ventaja competitiva.

Por ejemplo, las tecnologías IAM permiten que la empresa brinde a los usuarios fuera de la organización, por ejemplo, clientes, socios, contratistas y proveedores, acceso a su red a través de aplicaciones móviles, aplicaciones locales y aplicaciones de software como servicio sin comprometer la seguridad. Esto permite una mejor colaboración, mayor productividad, mayor eficiencia y costos operativos reducidos.

Los procesos de gestión del acceso a la identidad mal controlados pueden llevar a un incumplimiento regulatorio, ya que si se audita a la organización, la administración no podrá probar que los datos de la empresa no corren riesgo de abuso.

Los sistemas IAM ayudan a las empresas a cumplir mejor con las regulaciones gubernamentales al permitirles demostrar que la información corporativa no se está utilizando de manera incorrecta. Con las herramientas de gestión de identidad y acceso, las empresas también pueden demostrar que los datos necesarios para la auditoría pueden estar disponibles a pedido. [2] [12]

2.1.12 Librerías de java

Las librerías de java permiten realizar sistemas que protejan las credenciales de usuario gracias a los métodos que proporcionan. Estas librerías proporcionan distintos métodos y diferentes protocolos de seguridad para que se adapten a los distintos sistemas.

Los métodos que proporcionan estas librerías son desde la creación del usuario, el inicio de sesión del usuario, la creación de identidades de sesión, el uso de funciones de administrador. En resumen, proporcionan suficientes métodos para cubrir el ciclo de vida de un usuario en un sistema.

Existen infinitud de librerías, cómo pueden ser JJWT, ESAPI, JAAS… [13]

2.2 Resumen de soluciones

Para situarnos desde una perspectiva más global, en el apartado anterior hemos explicado de manera algo más concreta la evolución que se ha sufrido en la autenticación. Inicialmente eran métodos más básicos que únicamente, trataban de cifrar contraseñas.

Poco a poco, estos métodos se fueron haciendo más seguros con añadiendo por ejemplo, protocolos que securizasen también la comunicación de estos. Después de esto, apareciendo los SSO que pueden utilizar varios protocolos cómo hemos podido ver y permiten una autenticación en varios sistemas, con la misma identidad.

Después, comentamos los IAM, que como resumen, son Frameworks que permiten manejar el control de acceso y los permisos de los usuarios de manera sencilla. Esto facilita tanto a los programadores de los sistemas, que tienen que tener una menor preocupación en cuanto a la securización de la aplicación, ya que el IAM la proporciona y el administrador de la aplicación, que podrá manejar a los usuarios de manera más sencilla.

Por último, comentamos el uso de librerías Java, que nos permite una solución más sencilla y probablemente adaptable a un mayor número de sistemas. Por lo tanto, es posiblemente la solución válida para el objetivo de nuestro TFG.

3 Estadísticas de TFGs

Para tener una idea más clara del impacto que puede tener nuestro componente, se ha realizado un estudio con una muestra pequeña, para poder medir de manera aproximada cuantos TFG están verdaderamente protegidos. Los datos se han extraído de distintos TFG de informática desde e-archivo y son los siguientes:

	2017	2016	2015	2014	2013	Total
No protegen credenciales		3	1	2	1	7
Mencionan la ley, pero no indican el cómo la aplican		9	7	3	1	20
Protegen credenciales de usuario	2	1	2	6	3	14
No hay datos que proteger	1	18	9	8	2	38
Total	3	31	19	19	7	79

Tabla 1: Estadísticas de TFG

Cómo podemos observar, el número de TFGs que indican cómo protegen las credenciales y acreditamos cómo único válido, es un porcentaje muy pequeño de toda la muestra. Cómo podemos ver en la tabla anterior, de 79 TFG únicamente 14 protegen las credenciales correctamente.

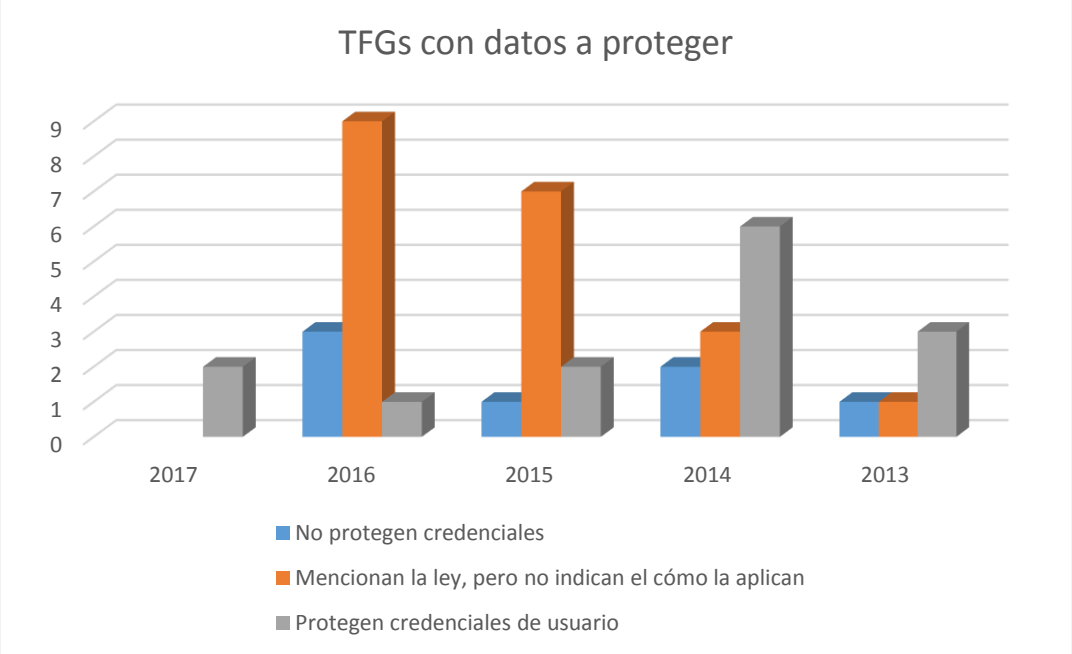

Ilustración 2: Estadísticas de TFGs con datos a proteger

Sí solo tenemos en cuenta los TFG que tienen datos que proteger, podemos ver que todos los años, los estudiantes tienen conocimiento de que se debe aplicar la ley de protección de datos. Sin embargo, un porcentaje muy pequeño explica de qué manera la aplica. Por ejemplo, con decir el cifrado que ha utilizado y los datos que ha protegido sería suficiente para saber que está aplicando correctamente la ley.

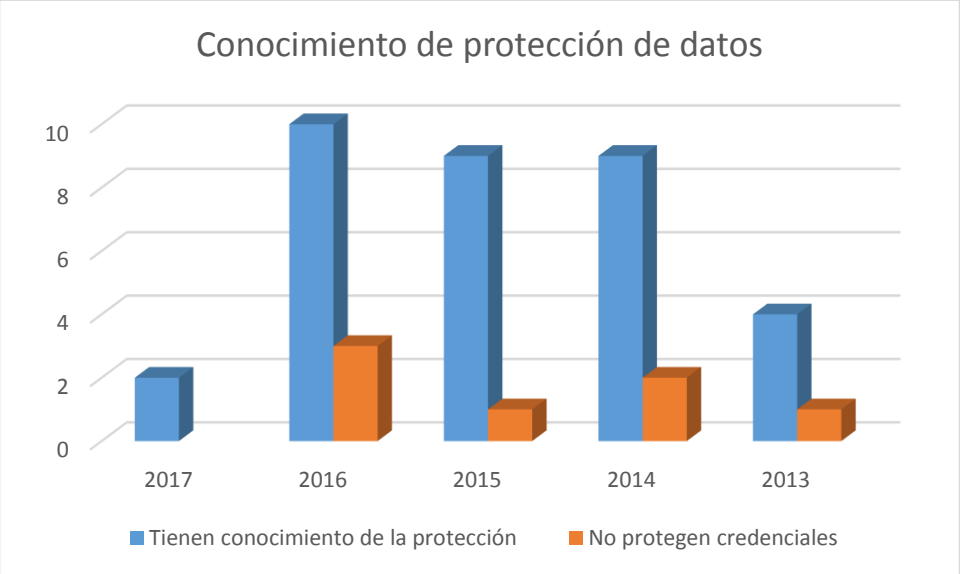

Ilustración 3: Estadísticas de conocimiento de protección de datos

Como punto positivo, podemos destacar que la mayoría de los estudiantes son conscientes de que hay que proteger los datos de los usuarios. Además, en algunos casos, puede haberse debido a un despiste que no mencionase la manera de proteger datos que utilizaban sus sistemas.

En uno de los anexos incluidos al finalizar el trabajo, se encuentra la colección de TFG analizados y su valoración.

4 Análisis

Para el desarrollo de componentes software es necesario analizar el problema que se quiere resolver para así, definir las necesidades que se deben satisfacer. Para ello, en este apartado se trataran diferentes aspectos cómo puede ser el ciclo de vida del software, los requisitos del sistema, la solución realizada y más aspectos relacionados con el análisis de software.

4.1 Ciclo de vida

Para el desarrollo del sistema, se ha elegido un ciclo de vida del software en cascada. Es el ciclo de vida del software más sencillo, pero se adapta a las necesidades del sistema que queremos realizar. Las fases se irán superando de manera secuencial y una vez se finalice, se podrá realizar una segunda fase. El nombre de desarrollo en cascada, es por las posiciones que ocupan las distintas fases en el esquema.

Se escoge este ciclo de vida ya que los requisitos del software a implementar son bastante claros y las necesidades que debe cubrir, no se espera que cambien en el tiempo que el proyecto esté en marcha. Se podrá ir mejorando la solución, pero esto se realiza en la propia fase de mantenimiento del software.

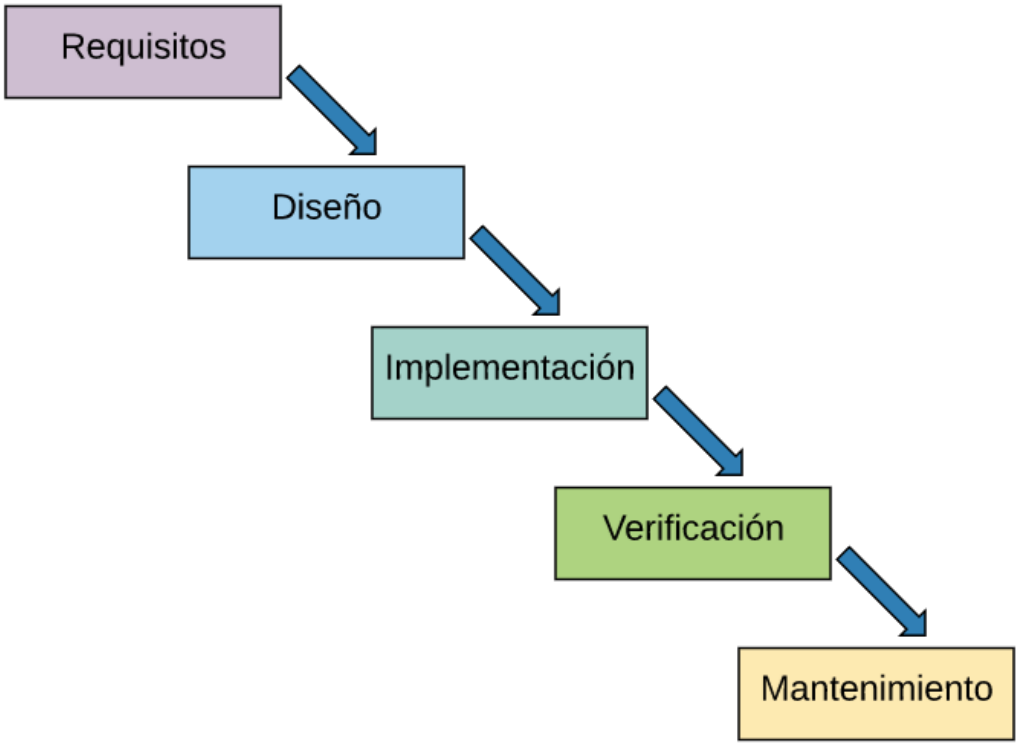

Ilustración 4: Modelo en cascada [14]

4.1.1 Fases del modelo

4.1.1.1 Requisitos del software

En esta fase se hace un análisis de las necesidades del cliente y a través de estas, generar los requisitos de software necesarios para que el sistema cumpla las necesidades

del cliente. Hay que ser especialmente cuidadoso en esta primera fase, ya que en este modelo no se pueden añadir nuevos requisitos en mitad del proceso de desarrollo.

Por lo tanto, es importante llegar a un acuerdo con el cliente ya que esta parte nos permite estimar de forma rigurosa las necesidades del software antes de su diseño. Además, permite tener una base a partir de la cual estimar el coste del producto, los riesgos y los plazos. [14]

4.1.1.2 Diseño

El diseño es la etapa en la que se le da forma al software, de manera interna y visual. Se desarrolla la estructura que tendrá el sistema, los elementos que lo componen y que hace cada una de estas partes, y la interacción entre ellos.

Existen dos tipos de diseño, el diseño a alto nivel y el diseño a bajo nivel. El diseño a alto nivel, se basa en describir a través de los datos obtenidos en el análisis el diseño del sistema a grandes rasgos. El diseño a bajo nivel, entra en un nivel de detalle superior, llegando a incluir incluso el código y la implementación del sistema. [14]

4.1.1.3 Implementación

En esta fase se desarrollara el código basándonos tanto en el análisis, cómo en el diseño. La programación es el proceso que lleva de la formulación de un problema de computación, a un programa que se ejecute produciendo los pasos necesarios para resolver dicho problema.

Al programar, tenemos que realizar actividades como el análisis de las condiciones, la creación de algoritmos, y su codificación en el lenguaje escogido. [14]

4.1.1.4 Verificación

Como su propio nombre indica, una vez se termina la fase de implementación se verifica que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos.

El objetivo de las pruebas es el de obtener información de la calidad del software, y sirven para: encontrar defectos o bugs, aumentar la calidad del software, refinar el código previamente escrito sin miedo a romperlo o introducir nuevos bugs, etc. [14]

4.1.1.5 Instalación y mantenimiento

Una vez se han desarrollado todas las funcionalidades del software y se ha comprobado que funcionan correctamente, se inicia la fase de instalación y mantenimiento. Se instala la aplicación en el sistema y se comprueba que funcione correctamente en el entorno en que se va a utilizar.

A partir de ahora hay que asegurarse de que el software funcione y hay que destinar recursos a mantenerlo. El mantenimiento del software consiste en la modificación del producto después de haber sido entregado al cliente, ya sea para corregir errores o para mejorar el rendimiento o las características.

El propósito de esta fase es mantener el valor del software a través del tiempo. Esto puede hacerse añadiendo nuevos requisitos, corrigiendo errores, renovando el aspecto visual, mejorando la eficiencia o añadiendo nueva tecnología. El periodo de mantenimiento puede durar años, por lo que es una fase clave del modelo en cascada.

Para llevar a cabo correctamente la fase de mantenimiento, se necesita trazar un plan de antemano que nos prepare para todos los escenarios que puedan producirse durante esta fase. Para evitar futuros conflictos con el cliente, en el plan hay que especificar cómo los usuarios solicitarán las modificaciones o la corrección de errores, hacer una estimación del coste de la modificación de funcionalidades o corrección de errores, quién se encargará del mantenimiento, durante cuánto tiempo se dará soporte al software, etc. [14]

4.2 Características de la solución proporcionada

Con el apoyo de la herramienta DIA, mostramos un pequeño diagrama a alto nivel del objetivo de nuestro módulo.

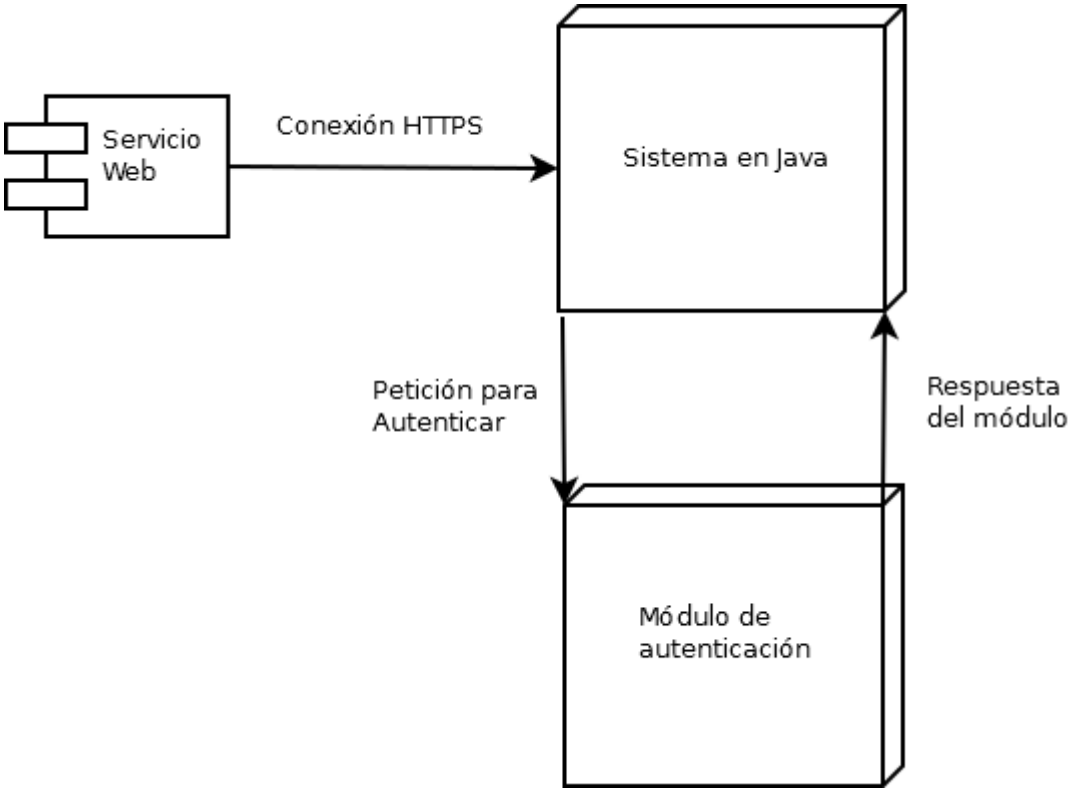

Ilustración 5: Diagrama del sistema con módulo de seguridad

Cómo podemos observar en la ilustración, el servicio web accedería a un sistema en java y en el momento tanto de crear usuarios, eliminarlos, autenticarlos se comunicaría con el módulo de autenticación. Es decir, cada vez que se realizan operaciones con los usuarios, se debe comunicar con este módulo. Este módulo de autenticación es el que desarrollamos en este proyecto.

El módulo de autenticación tiene que satisfacer una serie de características en base a lo comentado en el apartado Objetivos. Las características a cumplir son las siguientes.

- El sistema debe permitir registrar los usuarios en el sistema.
- El sistema debe permitir iniciar sesión a un usuario registrado en el sistema.
- El sistema debe permitir modificar la contraseña de los usuarios.

- El sistema debe permitir cerrar sesión a un usuario con sesión activa en el sistema.
- El sistema debe permitir eliminar un usuario del sistema.
- El sistema debe poder integrarse con facilidad en otros sistemas.
- El sistema debe cumplir con la regulación vigente de seguridad.
- El sistema debe permitir actualizar los protocolos de autenticación en caso de que estos queden obsoletos.
- El sistema debe tener un uso intuitivo y fácil para los usuarios.

4.3 Solución escogida

En el apartado Estudio de posibles soluciones, hicimos un estudio de los distintos métodos por los qué se puede proceder a la autenticación. Teniendo en cuenta que va a ser aplicado a sistemas que se realizan en TFG, se debe proporcionar una herramienta sencilla que pueda ser fácil de utilizar y que se pueda integrar en la mayor cantidad de estos.

Por lo tanto, vamos a realizar un módulo de autenticación en Java utilizando la librería ESAPI. La ESAPI es una colección gratis y abierta de todos los métodos de seguridad que un desarrollador necesita para construir una aplicación Web segura. [13]

La ESAPI es una librería desarrollada por la OWASP. Está permite la generación de un código WEB seguro. La librería proporciona métodos que permiten validaciones y controles para evitar diferentes ataques. A pesar de que en nuestro caso, está desarrollado en Java, se puede utilizar también por ejemplo en ASP.NET, PHP, ColdFusion, Javascript, Ruby y Python.

Algunas de las empresas que implementan ESAPI para la seguridad de sus sistemas web son por ejemplo American Express, Banco Mundial y US Navy.

La elección de esta herramienta, como mencionamos en el párrafo inicial de este punto, se debe a que debe ser un sistema de fácil implantación y adaptable a cualquier sistema. Las librerías de Java, nos aportaban estos beneficios, seleccionando ESAPI al ser la más completa que hemos revisado. [15]

Si hubiera que realizar un proyecto de más envergadura, lo recomendable sería utilizar un IAM que permite una mayor independencia y un mejor manejo de un mayor número de usuarios, utiliza mecanismos de seguridad que protegen las credenciales. Además permite monitorizar los accesos de manera más precisa, permitir acceso a usuarios externos al sistema. Un posible IAM puede ser Keycloak. [16]

4.4 Requisitos

A continuación se definen los requisitos necesarios para crear el módulo de seguridad. Inicialmente, se expondrán los requisitos que podemos extraer de la información que nos proporciona el usuario. Una vez analizados, se proporcionan los requisitos funcionales y no funcionales que formara parte del sistema.

4.4.1 Requisitos de usuario

Los requisitos de usuario definen la funcionalidad que el usuario nos proporciona sobre el software que quiere recibir. Estos requisitos, se dividen en dos grandes grupos, requisitos funcionales o de capacidad y no funcionales o de restricción.

- **Requisitos funcionales o de capacidad:** indican las funcionalidades del sistema y su comportamiento.
- **Requisitos no funcionales o de restricción:** son restricciones que limitan el sistema.

El formato que debe seguir todo requisito es el mostrado a continuación:

ID	RU-[Y]-[XXX]		
Título			
Descripción			
Versión		**Fecha**	
Fuente		**Estabilidad**	☐ Sí ☐ No
Prioridad	☐ Alta	☐ Media	☐ Baja
Necesidad	☐ Esencial	☐ Opcional	☐ Deseable
Claridad	☐ Alta	☐ Media	☐ Baja
Verificabilidad	☐ Alta	☐ Media	☐ Baja

Tabla 2: Formato de tabla de requisitos de usuario

A continuación definimos los atributos de la tabla:

- **ID:** Permite identificar de manera univoca cada uno de los requisitos, ya que cada requisito tiene un identificador único.
  - **RU:** Identificará al usuario como requisito de usuario.
  - **[Y]:** Indica el tipo de requisito de usuario que puede tener. Pueden ser dos valores:
    - **C:** Indicará que se trata de un requisito de capacidad.
    - **R:** Indicará que se trata de un requisito de restricción.
  - **[XXX]:** Valor que indica el número de requisito dentro de cada una de las categorías. Debe ser incremental, siendo 001 el primer valor y sin repetirse dentro de la misma categoría.

- **Título:** Proporciona una breve descripción de la funcionalidad.
- **Descripción:** Describe la funcionalidad o restricción que representa el requisito.
- **Versión:** Identifica la versión del requisito. Tomará valores numéricos positivos de 1 a N.
- **Fecha:** Indica la fecha en la que se ha añadido o modificado el requisito.
- **Fuente:** Indica el origen del requisito de usuario.
- **Estabilidad:** Indica si un requisito permanece estable en posteriores versiones.
    - **Si:** El requisito es estable.
    - **No:** El requisito no es estable.
- **Prioridad:** Indicará la importancia de un requisito.
    - **Alta:** El requisito es prioritario, deben ser los primeros en implementarse.
    - **Media:** El requisito es importante para el sistema. Debe implementarse cuando ya hayan sido implementado los requisitos de prioridad alta.
    - **Baja:** El requisito es deseable para el sistema.
- **Necesidad:** Evaluará la necesidad de un requisito para el sistema.
    - **Esencial:** El requisito es esencial para el correcto funcionamiento del sistema.
    - **Opcional:** El requisito es opcional para el funcionamiento del sistema.
    - **Deseable:** El requisito es deseable, pero no necesario para el sistema.
- **Claridad:** Evaluará la ambigüedad de un requisito.
    - **Alta:** El requisito carece de ambigüedad. Es claro y conciso.
    - **Media:** El requisito es claro, pero puede presentar problemas de ambigüedad de cara a la implementación.
    - **Baja:** El requisito es muy ambiguo y poco claro, presentando problemas para su correcta comprensión.
- **Verificabilidad:** Evaluará la verificabilidad de un requisito.
    - **Alta:** El requisito es verificable.
    - **Media:** El requisito puede ser dudoso de verificar.
    - **Baja:** El requisito es difícil de verificar.

4.4.1.1 Requisitos de usuario de capacidad

ID	RU-C-001		
Título	Almacenamiento de usuarios.		
Descripción	El sistema permitirá almacenar usuarios de manera segura. Además permitirá almacenar los cambios que se realicen en estos.		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 3: Requisito de usuario RU-C-001

ID	RU-C-002		
Título	Registro de usuario.		
Descripción	El sistema permitirá registrar usuarios. Para ello, deberá introducir un usuario y una contraseña.		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 4: Requisito de usuario RU-C-002

ID	RU-C-003		
Título	Modificar contraseña.		
Descripción	El sistema permitirá modificar la contraseña al usuario. Será necesario que el usuario introduzca la contraseña antigua para poder introducir una nueva.		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 5: Requisito de usuario RU-C-003

ID	RU-C-004		
Título	Inicio de sesión.		
Descripción	El sistema permitirá iniciar sesión al usuario. Para ello, el usuario deberá introducir el usuario y la contraseña.		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 6: Requisito de usuario RU-C-004

ID	RU-C-005		
Título	Cerrar sesión.		
Descripción	El sistema permitirá cerrar sesión al usuario.		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 7: Requisito de usuario RU-C-005

ID	RU-C-006		
Título	Borrar usuario.		
Descripción	El sistema permitirá al usuario borrar su usuario.		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 8: Requisito de usuario RU-C-006

ID	RU-C-007		
Título	Usuario administrador.		
Descripción	El sistema permitirá proporcionara un usuario administrador que podrá consultar los usuarios y reestablecer contraseñas en caso de petición del usuario.		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☒ Media	☐ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 9: Requisito de usuario RU-C-007

4.4.1.2 Requisitos de restricción

ID	RU-R-001		
Título	Contraseña segura.		
Descripción	El sistema establecerá un formato de contraseña seguro. Por lo tanto, la contraseña deberá tener las siguientes características: • Deberá tener una letra mayúscula • Deberá tener un valor numérico		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 10: Requisito de usuario RU-R-001

ID	RU-R-002		
Título	Usuario administrador único.		
Descripción	El sistema permitirá un único administrador, pudiendo realizar todas las funciones del administrador		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☒ Media	☐ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 11: Requisito de usuario RU-R-002

ID	RU-R-003		
Título	Visualización de usuarios.		
Descripción	El sistema permitirá únicamente al usuario administrador ver los usuarios registrados en el sistema.		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☒ Media	☐ Baja
Necesidad	☐ Esencial	☐ Opcional	☒ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 12: Requisito de usuario RU-R-003

ID	RU-R-004		
Título	Protección de los datos de usuario.		
Descripción	La aplicación web aplicara la ley de protección de datos a todos los datos personales.		
Versión	1	**Fecha**	07/05/2018
Fuente	Usuario	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 13: Requisito de usuario RU-R-004

4.4.2 Requisitos software.

Los requisitos software permiten una descripción completa de la funcionalidad del sistema a través de los requisitos de usuario. A continuación, se describen los requisitos de software, los cuales, al igual que los requisitos de usuario, tienen dos tipos de requisitos.

- Requisitos funcionales: Indican las funcionalidades que el sistema debe proporcionar.
- Requisitos no funcionales. Restricciones que afectan a los servicios o funciones del sistema.

La plantilla que se utilizará para los requisitos software es la siguiente.

ID	RS-[Y][XXX]		
Título			
Descripción			
Versión		**Fecha**	
Referencia		**Estabilidad**	☐ Sí ☐ No
Prioridad	☐ Alta	☐ Media	☐ Baja
Necesidad	☐ Esencial	☐ Opcional	☐ Deseable
Claridad	☐ Alta	☐ Media	☐ Baja
Verificabilidad	☐ Alta	☐ Media	☐ Baja

Tabla 14: Plantilla de tabla de requisitos software.

A continuación definimos los atributos de la tabla:

- **ID:** Permite identificar de manera univoca cada uno de los requisitos, ya que cada requisito tiene un identificador único.
  - **RS:** Identificará al usuario como requisito de software.
  - **[Y]:** Puede tomar diez posibles valores.
    - **F:** Indicará que se trata de un requisito funcional.
    - **NF:** Indicará que se trata de un requisito funcional.
  - **[XXX]:** Valor que indica el número de requisito dentro de cada una de las categorías. Debe ser incremental, siendo 001 el primer valor y sin repetirse dentro de la misma categoría.
- **Título:** Proporciona una breve descripción de la funcionalidad.
- **Descripción:** Describe la funcionalidad o restricción que representa el requisito.
- **Versión:** Identifica la versión del requisito. Tomará valores numéricos positivos de 1 a N.
- **Fecha:** Indica la fecha en la que se ha añadido o modificado el requisito.

- **Referencia:** Indica con que requisito de usuario tiene relación el requisito de software.
- **Estabilidad:** Indica si un requisito permanece estable en posteriores versiones.
    - **Si:** El requisito es estable.
    - **No:** El requisito no es estable.
- **Prioridad:** Indicará la importancia de un requisito.
    - **Alta:** El requisito es prioritario, deben ser los primeros en implementarse.
    - **Media:** El requisito es importante para el sistema. Debe implementarse cuando ya hayan sido implementado los requisitos de prioridad alta.
    - **Baja:** El requisito es deseable para el sistema.
- **Necesidad:** Evaluará la necesidad de un requisito para el sistema.
    - **Esencial:** El requisito es esencial para el correcto funcionamiento del sistema.
    - **Opcional:** El requisito es opcional para el funcionamiento del sistema.
    - **Deseable:** El requisito es deseable, pero no necesario para el sistema.
- **Claridad:** Evaluará la ambigüedad de un requisito.
    - **Alta:** El requisito carece de ambigüedad. Es claro y conciso.
    - **Media:** El requisito es claro, pero puede presentar problemas de ambigüedad de cara a la implementación.
    - **Baja:** El requisito es muy ambiguo y poco claro, presentando problemas para su correcta comprensión.
- **Verificabilidad:** Evaluará la verificabilidad de un requisito.
    - **Alta:** El requisito es verificable.
    - **Media:** El requisito puede ser dudoso de verificar.
    - **Baja:** El requisito es difícil de verificar.

4.4.2.1 Requisitos funcionales

ID	RS-F001		
Título	Introducir usuarios		
Descripción	El sistema permitirá insertar los datos de usuario. Estos datos se almacenaran en un fichero. Estarán formados por: • Id de usuario. • Nombre del usuario. • Contraseña (Función resumen). • Roles. • Bloqueado. • Habilitado. • Contraseñas antiguas. • Último Login. • Último Login fallado. • Tiempo de expiración. • Intentos de Login fallidos.		
Versión	2	Fecha	05/07/2018
Referencia	RU-C-001	Estabilidad	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 15: Requisito software RS-F001

ID	RS-F002		
Título	Almacenar modificaciones		
Descripción	El sistema actualizara la información del fichero si se realiza alguna modificación o borrado en el usuario.		
Versión	2	Fecha	05/07/2018
Referencia	RU-C-001	Estabilidad	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 16: Requisito software RS-F002

ID	RS-F003		
Título	Crear usuarios		
Descripción	El sistema permitirá crear nuevos usuarios a través de un formulario que estará compuesto por: • Usuario. • Contraseña. • Repetir contraseña.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-C-002	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 17: Requisito software RS-F003

ID	RS-F004		
Título	Modificar contraseña.		
Descripción	El sistema permitirá modificar la contraseña del usuario a través de un formulario compuesto por: • Usuario. • Contraseña antigua. • Nueva Contraseña. • Repetir nueva contraseña.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-C-003	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 18: Requisito software RS-F004

ID	RS-F005		
Título	Iniciar sesión en el sistema.		
Descripción	El sistema permitirá iniciar sesión al usuario cuando esté introduzca las credenciales de manera correcta. Las credenciales estarán compuestas por: • Usuario. • Contraseña.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-C-004	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 19: Requisito software RS-F005

ID	RS-F006		
Título	Cerrar sesión.		
Descripción	El sistema permitirá cerrar sesión al usuario si este tiene una sesión activa en el sistema.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-C-005	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 20: Requisito software RS-F006

ID	RS-F007		
Título	Borrar usuario del sistema.		
Descripción	El sistema permitirá a los usuarios eliminar su cuenta sí están registrados.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-C-006	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☒ Media	☐ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☐ Alta	☒ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 21: Requisito software RS-F007

ID	RS-F008		
Título	Usuario administrador.		
Descripción	El sistema proporcionara un administrador creado por defecto.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-C-007	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☒ Media	☐ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☐ Alta	☒ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 22: Requisito software RS-F008

ID	RS-F009		
Título	Consulta de usuarios.		
Descripción	El sistema permitirá que el usuario administrador pueda consultar los datos almacenados de los usuarios del sistema. • Usuario actual. • Ultimo acceso con éxito. • Último acceso sin éxito. • Intentos erróneos actuales. • Roles. • Último host. • Cookies. • Cookies del navegador.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-C-007	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☐ Media	☒ Baja
Necesidad	☐ Esencial	☐ Opcional	☒ Deseable
Claridad	☐ Alta	☒ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 23: Requisito software RS-F009

ID	RS-F010		
Título	Reseteo de contraseña.		
Descripción	El sistema permitirá que el usuario administrador pueda resetear la contraseña a un usuario.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-C-007	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☐ Media	☒ Baja
Necesidad	☐ Esencial	☐ Opcional	☒ Deseable
Claridad	☐ Alta	☒ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 24: Requisito software RS-F010

ID	RS-F011		
Título	Eliminar usuario.		
Descripción	El sistema permitirá que el usuario administrador pueda eliminar un usuario del sistema.		
Versión	1	Fecha	24/05/2018
Referencia	RU-C-007	Estabilidad	☒ Sí ☐ No
Prioridad	☐ Alta	☐ Media	☒ Baja
Necesidad	☐ Esencial	☐ Opcional	☒ Deseable
Claridad	☐ Alta	☒ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 25: Requisito software RS-F011

4.4.2.2 Requisitos no funcionales

ID	RS-NF001		
Título	Formato de contraseña.		
Descripción	El sistema establecerá un formato de contraseña en el que se deben cumplir tres de las siguiente características y un tamaño de 8 carácteres: • Deberá tener una letra minúscula. • Deberá tener una letra mayúscula. • Deberá tener un valor numérico. • Deberá tener un carácter especial.		
Versión	1	Fecha	24/05/2018
Referencia	RU-R-001	Estabilidad	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 26: Requisito software RS-NF001

ID	RS-NF002		
Título	Protección de credenciales.		
Descripción	El sistema proporcionará un mecanismo que envíe y almacene las credenciales de usuario de manera segura.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-R-001	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 27: Requisito software RS-NF002

ID	RS-NF003		
Título	Contraseña invalida.		
Descripción	El sistema bloqueará la cuenta del usuario durante un tiempo variable en caso de introducir unas credenciales erróneas en 3 ocasiones consecutivas.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-R-001	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☐ Media	☒ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☐ Alta	☒ Media	☐ Baja
Verificabilidad	☐ Alta	☒ Media	☐ Baja

Tabla 28: Requisito software RS-NF003

ID	RS-NF004		
Título	Contraseña invalida.		
Descripción	El sistema no permitirá cambiar la contraseña a una nueva si esta no cumple requisitos de seguridad.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-R-001	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☐ Media	☒ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☐ Alta	☒ Media	☐ Baja
Verificabilidad	☐ Alta	☒ Media	☐ Baja

Tabla 29: Requisito software RS-NF004

ID	RS-NF005		
Título	Limitación de administradores.		
Descripción	El sistema permitirá únicamente un usuario administrador.		
Versión	1	Fecha	24/05/2018
Referencia	RU-R-002	Estabilidad	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 30: Requisito software RS-NF005

ID	RS-NF006		
Título	Funciones del administrador.		
Descripción	El sistema permitirá únicamente al usuario administrador reestablecer contraseñas en caso de pérdida.		
Versión	1	Fecha	24/05/2018
Referencia	RU-R-002	Estabilidad	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☐ Alta	☒ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 31: Requisito software RS-NF006

ID	RS-NF007		
Título	Visualizar usuarios.		
Descripción	El sistema permitirá únicamente visualizar todos los usuarios al usuario administrador del sistema.		
Versión	1	Fecha	24/05/2018
Referencia	RU-R-003	Estabilidad	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☐ Alta	☒ Media	☐ Baja
Verificabilidad	☐ Alta	☒ Media	☐ Baja

Tabla 32: Requisito software RS-NF007

ID	RS-NF008		
Título	Ley de Protección de datos.		
Descripción	El sistema cumplirá la ley de protección de datos vigente.		
Versión	1	**Fecha**	24/05/2018
Referencia	RU-R-004	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 33: Requisito software RS-NF008

ID	RS-NF009		
Título	Notificación de errores.		
Descripción	El sistema mostrará los errores en caso de existir.		
Versión	1	**Fecha**	24/05/2018
Referencia	Analista	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☐ Alta	☒ Media	☐ Baja
Verificabilidad	☐ Alta	☒ Media	☐ Baja

Tabla 34: Requisito software RS-NF009

ID	RS-NF010		
Título	Secuencia de pasos.		
Descripción	El sistema permitirá acceder a cualquier funcionalidad en menos de 5 pasos.		
Versión	1	**Fecha**	24/05/2018
Referencia	Analista	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☒ Media	☐ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 35: Requisito software RS-NF010

ID	RS-NF011		
Título	Acceso al fichero.		
Descripción	El sistema realizara consultas, inserciones y borrados en el fichero de almacenamiento.		
Versión	1	**Fecha**	24/05/2018
Referencia	Analista	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☒ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 36: Requisito software RS-NF011

ID	RS-NF012		
Título	Idioma del sistema.		
Descripción	El sistema mostrará todas sus funcionalidades en español.		
Versión	1	**Fecha**	24/05/2018
Referencia	Analista	**Estabilidad**	☒ Sí ☐ No
Prioridad	☐ Alta	☒ Media	☐ Baja
Necesidad	☐ Esencial	☒ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 37: Requisito software RS-NF012

ID	RS-NF013		
Título	Implantación del sistema.		
Descripción	El sistema deberá poder ser implantado en cualquier sistema como un módulo independiente.		
Versión	1	**Fecha**	24/05/2018
Referencia	Analista	**Estabilidad**	☒ Sí ☐ No
Prioridad	☒ Alta	☐ Media	☐ Baja
Necesidad	☒ Esencial	☐ Opcional	☐ Deseable
Claridad	☒ Alta	☐ Media	☐ Baja
Verificabilidad	☒ Alta	☐ Media	☐ Baja

Tabla 38: Requisito software RS-NF013

4.5 Matriz de trazabilidad.

Las matrices de trazabilidad relacionan dos elementos esenciales para comprobar que el desarrollo del software se está realizando correctamente. Es decir, es una herramienta clave para la ingeniería de los proyectos, así como para el seguimiento de los diversos elementos que los componen.

En este caso, a continuación, exponemos la matriz de trazabilidad entre los requisitos de usuario, y los requisitos software.

	RU-C-001	RU-C-002	RU-C-003	RU-C-004	RU-C-005	RU-C-006	RU-C-007	RU-R-001	RU-R-002	RU-R-003	RU-R-004
RS-F001	X										
RS-F002	X										
RS-F003		X									
RS-F004			X								
RS-F005				X							
RS-F006					X						
RS-F007						X					
RS-F008							X				
RS-F009							X				
RS-F010							X				
RS-F011							X				
RS-NF001								X			
RS-NF002								X			
RS-NF003								X			
RS-NF004								X			
RS-NF005									X		
RS-NF006									X		
RS-NF007										X	
RS-NF008											X
RS-NF009											
RS-NF010											
RS-NF011											
RS-NF012											
RS-NF013											

Tabla 39: Matriz de trazabilidad RU vs RS

4.6 Entorno operacional.

En este proyecto, el entorno operacional en el que se va a implementar el sistema no está definido, ya que el objetivo de este, es que pueda ser utilizado en el mayor número de sistemas posibles. Por lo tanto, ponemos unos requisitos mínimos basándonos en los rendimientos actuales de los sistemas.

Requisitos mínimos	
Sistema Operativo	Windows 8
Memoria RAM	2GB
Disco duro	100GB

Tabla 40: Requisitos mínimos para la implantación del sistema

Sin embargo, para el desarrollo del sistema, se realiza desde un ordenador personal. Para ello, se ha utilizado el IDE Eclipse, y las características del sistema son las siguientes:

Entorno de desarrollo	
Sistema Operativo	Windows 8
Memoria RAM	8GB
Disco duro	500GB

Tabla 41: Características del entorno de desarrollo

5 Diseño

A continuación, se explican las diferentes decisiones de diseño que se han tomado a la hora de realizar el sistema. Se definirán características cómo puede ser el tipo de arquitectura, los casos de uso del sistema, diagramas de secuencia y diseño de la interfaz.

Comenzaremos definiendo el tipo de arquitectura escogida, y tras esto, continuaremos con los casos de uso y su definición mediante esquemas y tablas, y por último, el diseño de datos y de interfaz.

5.1 Tipo de arquitectura.

La arquitectura escogida para realizar la aplicación ha sido una arquitectura Modelo-Vista-Controlador. El modelo-vista-controlador, es una filosofía de diseño de aplicaciones que consta de tres partes:

- Modelo:
  - Contiene la información del sistema.
  - Contiene la lógica de negocio.
  - Extrae los datos que requiere la vista.
- Vista:
  - Es la parte que percibe el usuario de la aplicación.
  - Realiza las peticiones al modelo y las representa de manera visual.
- Controlador:
  - Interacciona entre la vista y controlador permitiendo el correcto funcionamiento del sistema.

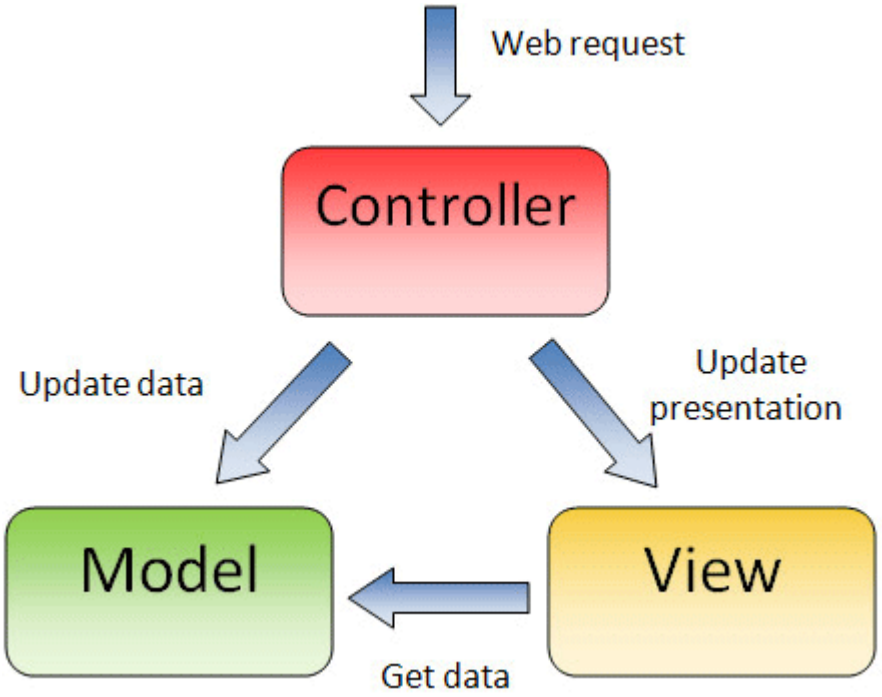

Ilustración 6: Modelo vista controlador. [17]

Se escoge ya que el sistema va a tener una interfaz que facilitara al usuario el uso de la aplicación, un lugar para almacenar los datos y por último los diferentes métodos necesarios para el manejo de los datos.

En el sistema, la vista está formada por los JSP que crearan la interfaz del usuario. El modelo está formado por el fichero en el que almacena los datos y las funciones de sus modificaciones. Por último, el controlador manejara las funciones que se tienen que realizar para el tratamiento de datos. Por lo tanto, estará formado tanto por el controlador (Servlet) y métodos auxiliares que permiten el correcto funcionamiento.

5.2 Casos de uso y diagramas de secuencia

En este apartado, explicamos los casos de uso, que son las distintas acciones que puede realizar el usuario del sistema. En ellos, se detallaran los pasos a seguir para realizar las acciones. Estas acciones se ven definidas por los requisitos del sistema tomados en el análisis del sistema.

Para cada caso de uso del sistema, se utilizará la siguiente plantilla.

CU – XX				
Título				
Objetivo				
Pre-condiciones				
Post-condiciones				
Escenario	**1**			
	2			
	.			
	.			
	N			
Versión			**Fecha**	
Actor				

Tabla 42: Plantilla de casos de uso

La tabla de casos de uso está formada por los siguientes elementos:

- **Identificador:** Identificará cada caso de uso de manera univoca. Se localiza en la parte superior de la tabla. Su formato será CU-XX, siendo XX un número de dos cifras incremental desde el valor 01.
- **Objetivo:** Define la acción que pueden realizar los actores con el sistema.
- **Pre-condiciones:** Son las condiciones necesarias para poder realizar la acción.
- **Post-condiciones:** Es el estado en el que queda el sistema una vez se ha realizado la acción (si no se ha producido ningún error).
- **Escenario:** Conjunto de pasos que deberá realizar el actor para conseguir el objetivo definido.

- **Versión:** Se introducirá la versión del caso de uso, para evitar errores con las versiones.
- **Fecha:** Fecha de realización del caso de uso.
- **Actor:** Será el tipo de usuario que utiliza la aplicación.

Para completar los casos de uso, se incluirá un diagrama UML de secuencia donde se podrá observar una descripción del escenario. Es decir, se indicará la interacción con las distintas partes del sistema en consonancia con cada caso de uso. Para un mismo caso de uso, podrán existir varios diagramas de secuencia. Los diagramas de secuencia han sido generados usando la herramienta DIA.

A continuación, mostramos una ilustración global de los casos de uso del sistema y sus actores y después cada uno individualmente de manera detallada.

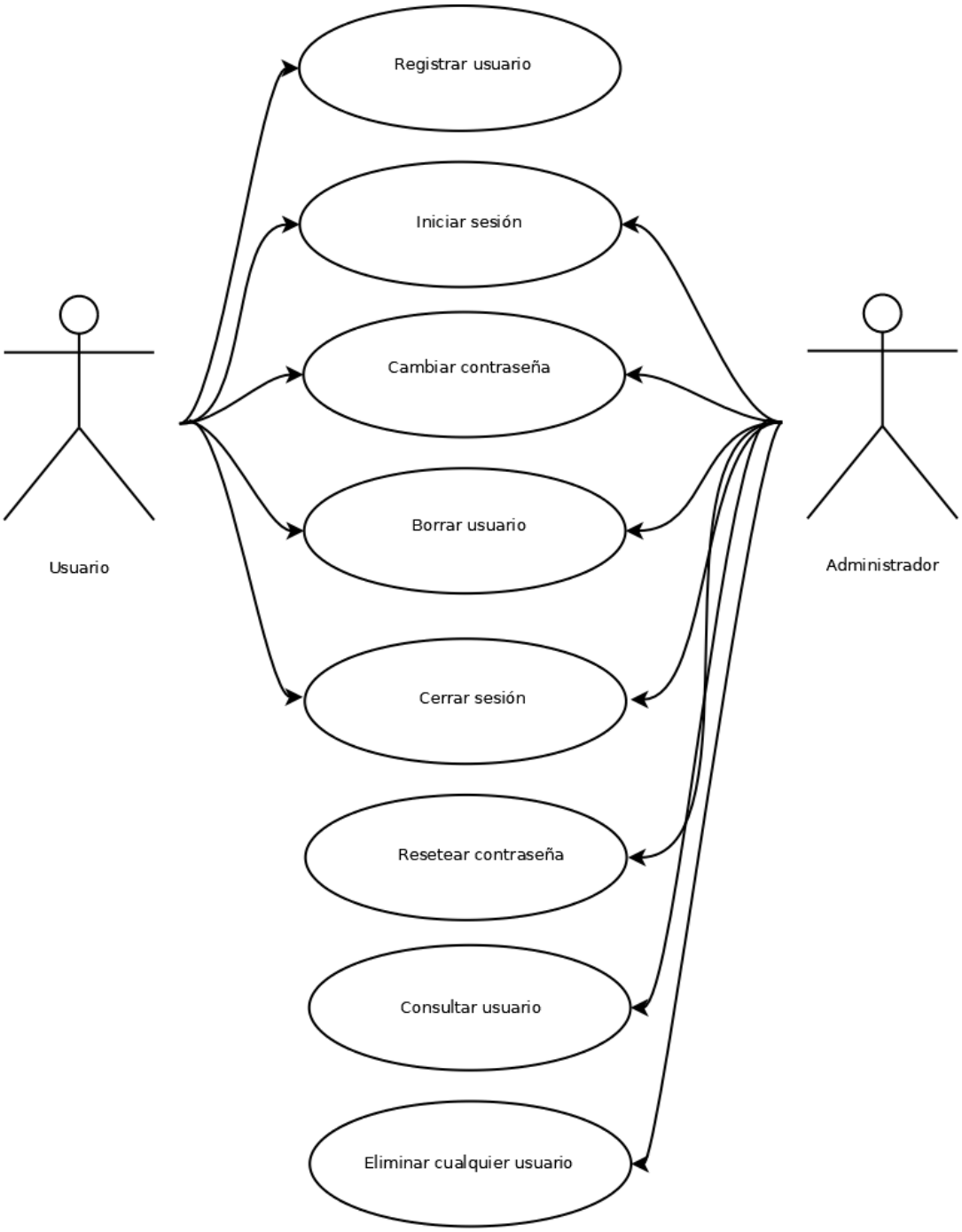

Ilustración 7: Casos de uso del sistema

CU – 01				
Título	Crear Usuario			
Objetivo	Los usuarios podrán crear un usuario en el sistema a través de un formulario.			
Pre-condiciones	• Conexión al servidor			
Post-condiciones	El usuario quedará registrado en el sistema			
Escenario	**1**	Cargar la página de inicio del sistema		
	2	Pulsar en el botón “Registrar Usuario”		
	3	Rellenar los siguientes datos: • Nombre de usuario. • Contraseña. • Confirmación de contraseña.		
	4	Pulsar en registrar.		
Versión	1		**Fecha**	10/06/2018
Actor	Usuario.			

Tabla 43: Caso de uso CU-01

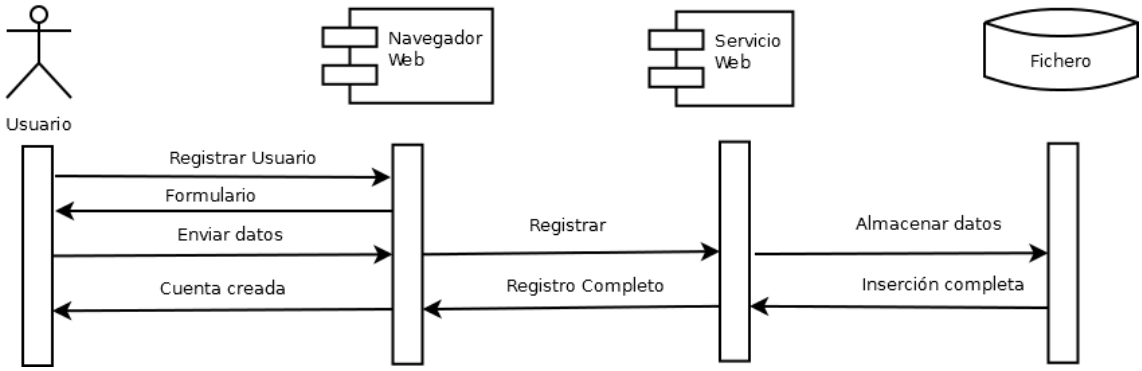

Ilustración 8: Diagrama Secuencia CU-01 (1)

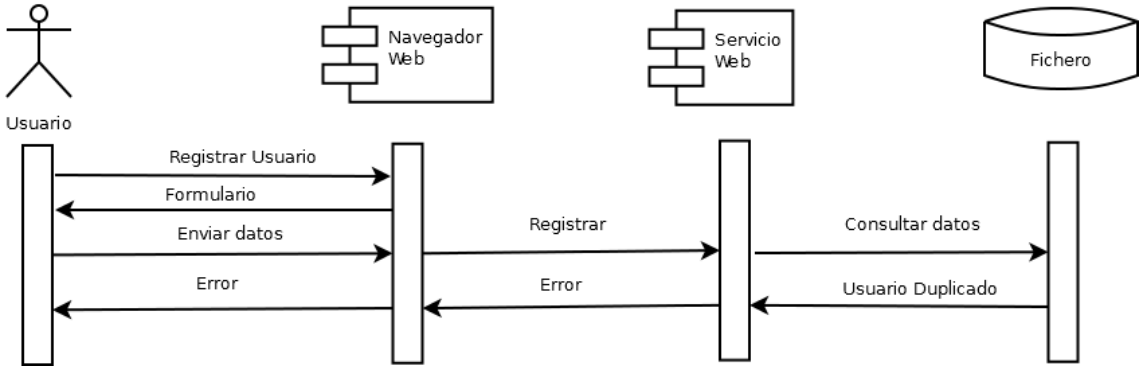

Ilustración 9: Diagrama Secuencia CU-01 (2)

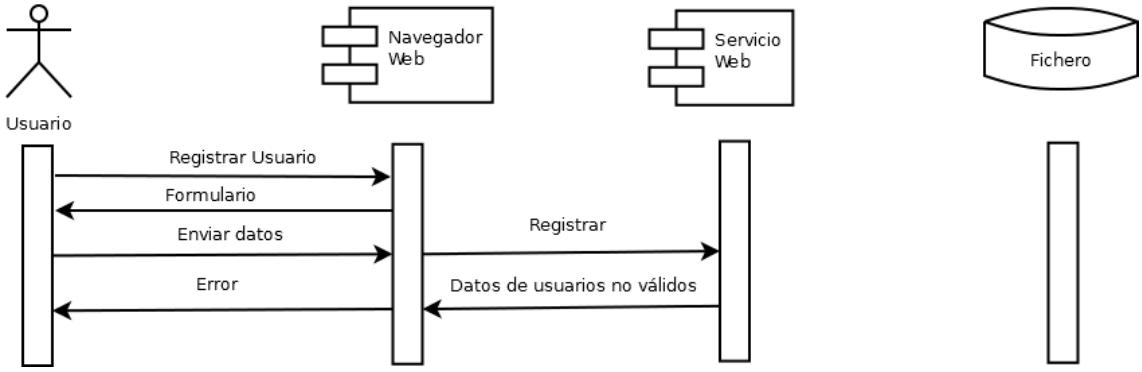

Ilustración 10: Diagrama Secuencia CU-01 (3)

CU - 02			
Título	Iniciar Sesión		
Objetivo	Los usuarios podrán crear un iniciar sesión en el sistema a través de un formulario.		
Pre-condiciones	• Conexión al servidor. • Usuario creado en el sistema.		
Post-condiciones	El usuario quedara con sesión activa en el sistema		
Escenario	**1**	Cargar la página de inicio del sistema	
	2	Rellenar los siguientes datos: • Usuario. • Contraseña.	
	3	Pulsar en Login.	
Versión	1	**Fecha**	10/06/2018
Actor	Usuario y Administrador.		

Tabla 44: Caso de uso CU-02

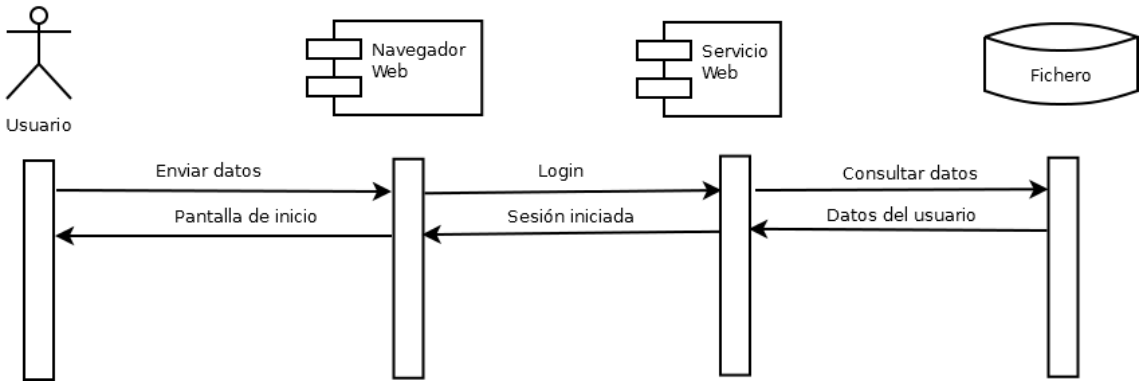

Ilustración 11: Diagrama Secuencia CU-02 (1)

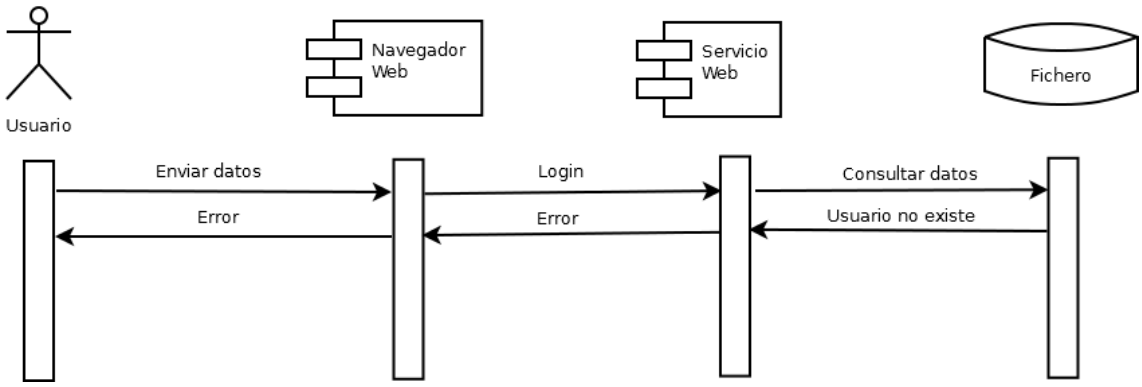

Ilustración 12: Diagrama Secuencia CU-02 (2)

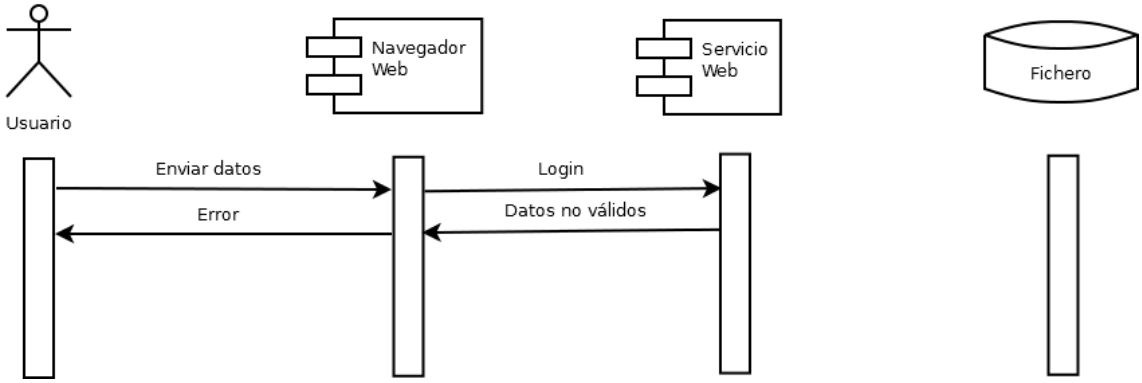

Ilustración 13: Diagrama Secuencia CU-02 (3)

CU - 03			
Título	Cambiar contraseña		
Objetivo	Los usuarios podrán modificar la contraseña en el sistema a través de un formulario.		
Pre-condiciones	• Conexión al servidor. • Usuario creado en el sistema.		
Post-condiciones	El usuario será modificado en el sistema.		
Escenario	**1**	Pulsar en cambiar contraseña	
	2	Rellenar los siguientes datos: • Usuario. • Antigua Contraseña. • Nueva Contraseña. • Repite nueva Contraseña	
	3	Pulsar en Guardar.	
Versión	1	**Fecha**	10/06/2018
Actor	Usuario y Administrador		

Tabla 45: Caso de uso CU-03

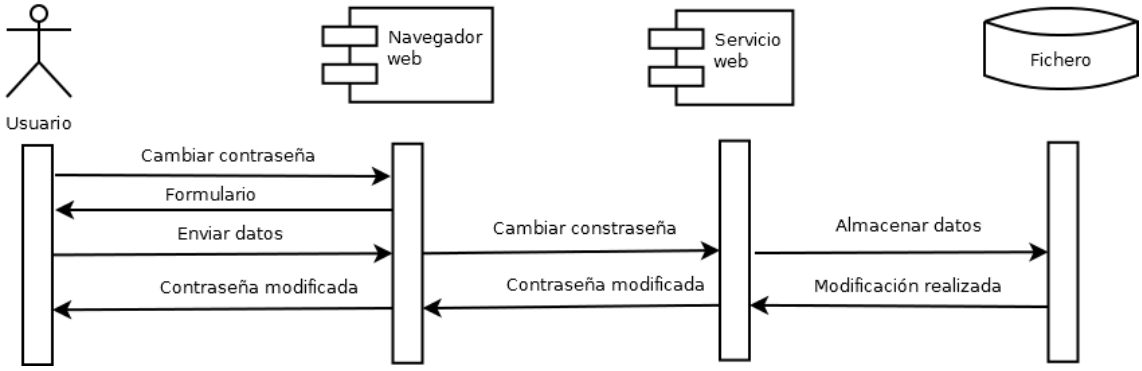

Ilustración 14: Diagrama Secuencia CU-03 (1)

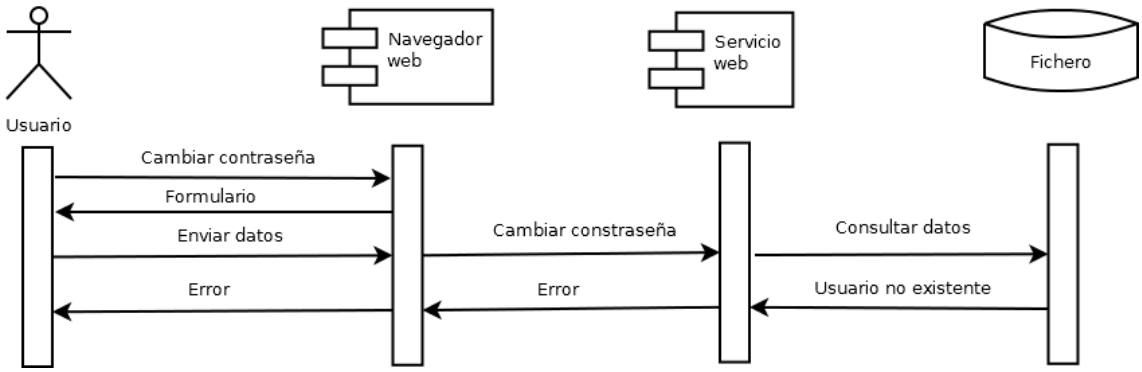

Ilustración 15: Diagrama Secuencia CU-03 (2)

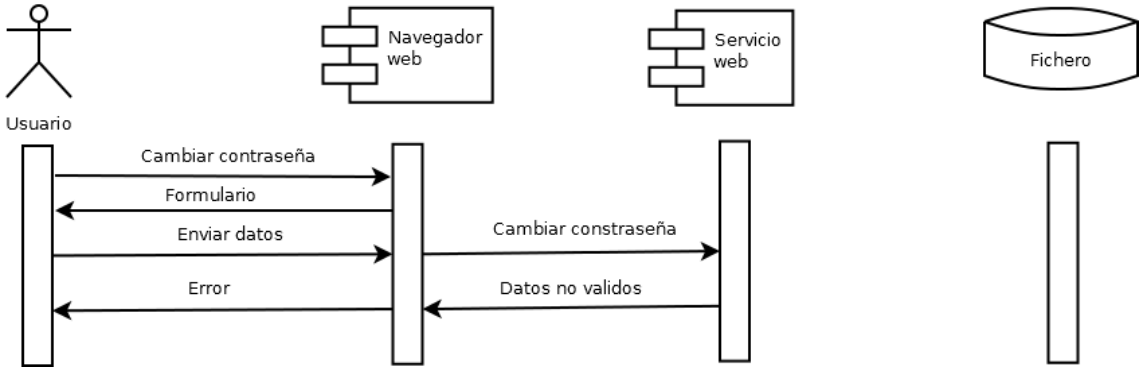

Ilustración 16: Diagrama Secuencia CU-03 (3)

CU - 04			
Título	Borrar usuario		
Objetivo	Los usuarios podrán borrarse del sistema a través de un botón.		
Pre-condiciones	• Conexión al servidor. • Usuario creado en el sistema. • Usuario logado.		
Post-condiciones	El usuario será eliminado del sistema.		
Escenario	**1**	Ir a la pantalla inicial tras el login	
	2	Pulsar el botón Borrar Usuario.	
Versión	1	**Fecha**	10/06/2018
Actor	Usuario y Administrador		

Tabla 46: Caso de uso CU-04

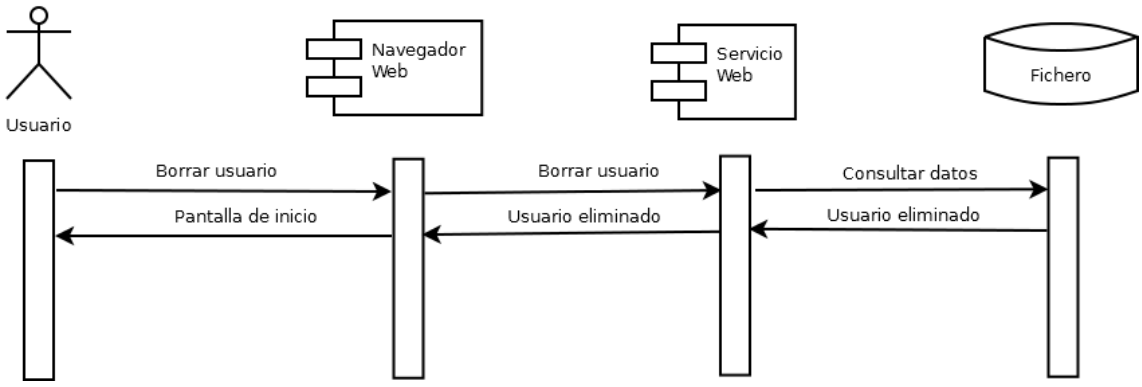

Ilustración 17: Diagrama Secuencia CU-04

CU – 05				
Título	Cerrar Sesión			
Objetivo	Los usuarios podrán cerrar la sesión en el sistema.			
Pre-condiciones	• Conexión al servidor. • Usuario creado en el sistema. • Usuario logado.			
Post-condiciones	El usuario se desconectará de la sesión.			
Escenario	**1**	Ir a la pantalla inicial tras el login		
	2	Pulsar el botón cerrar sesión.		
Versión	1		**Fecha**	10/06/2018
Actor	Usuario y Administrador			

Tabla 47: Caso de uso CU-05

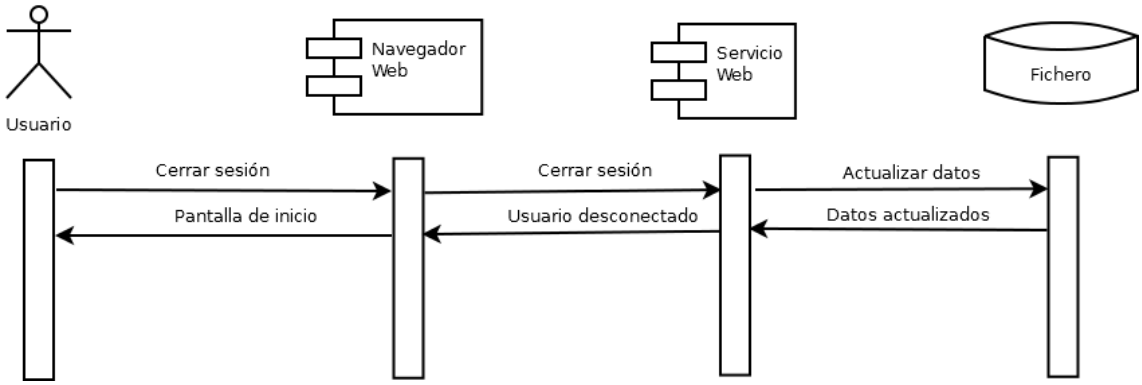

Ilustración 18: Diagrama Secuencia CU-05

CU – 06			
Título	Resetear Contraseña		
Objetivo	El administrador del sistema podrá resetear la contraseña de cualquiera de los empleados.		
Pre-condiciones	• Conexión al servidor. • Usuario creado en el sistema. • Usuario logado. • Usuario con rol administrador.		
Post-condiciones	El usuario que quiere resetear la contraseña será modificado.		
Escenario	**1**	Ir a la pantalla inicial tras el login	
	2	Pulsar el botón administrador.	
	3	Pulsar el botón resetear contraseña.	
	4	Rellenar el formulario de cambio de contraseña: • Usuario. • Nueva contraseña. • Confirmación de nueva contraseña.	
	5	Pulsar en guardar.	
Versión	1	**Fecha**	10/06/2018
Actor	Administrador		

Tabla 48: Caso de uso CU-06

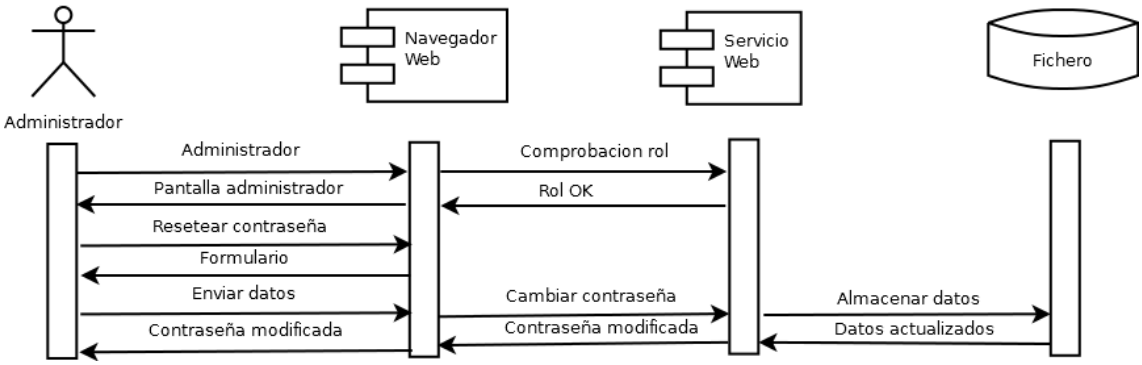

Ilustración 19: Diagrama Secuencia CU-06 (1)

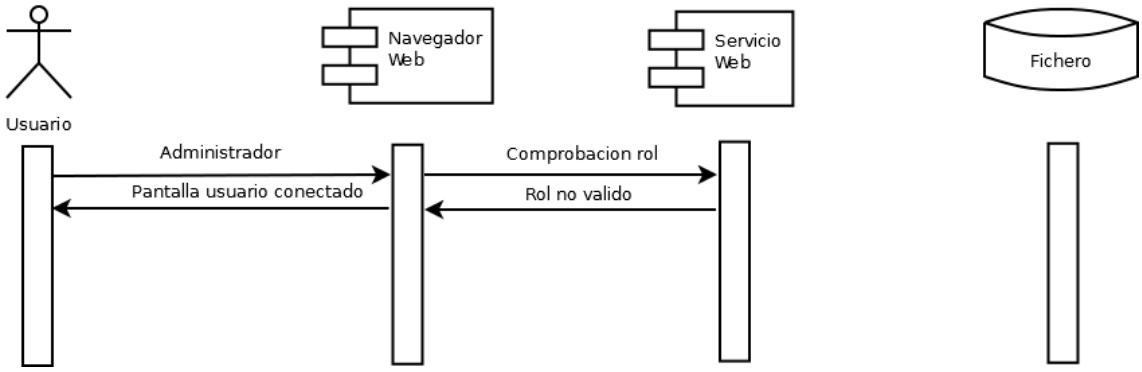

Ilustración 20: Diagrama Secuencia CU-06 (2)

CU – 07			
Título	Consultar usuario		
Objetivo	El administrador del sistema podrá consultar los datos de cualquiera de los empleados.		
Pre-condiciones	• Conexión al servidor. • Usuario creado en el sistema. • Usuario logado. • Usuario con rol administrador.		
Post-condiciones	El sistema mostrará la información del usuario a consultar.		
Escenario	**1**	Ir a la pantalla inicial tras el login	
	2	Pulsar el botón administrador.	
	3	Pulsar el botón consultar usuario.	
	4	Introducir en el formulario el campo: • Usuario.	
	5	Pulsar en consultar.	
	6	Mostrará la información del usuario en la pantalla.	
Versión	1	**Fecha**	10/06/2018
Actor	Administrador		

Tabla 49: Caso de uso CU-07

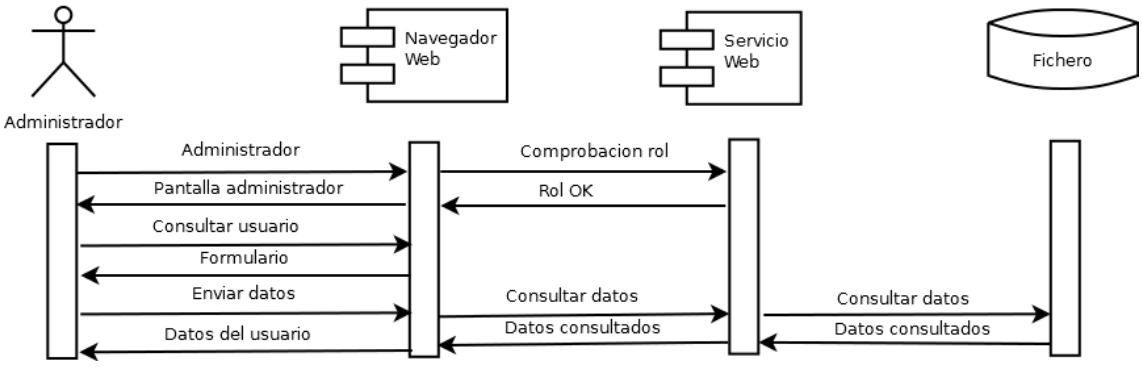

Ilustración 21: Diagrama Secuencia CU-07 (1)

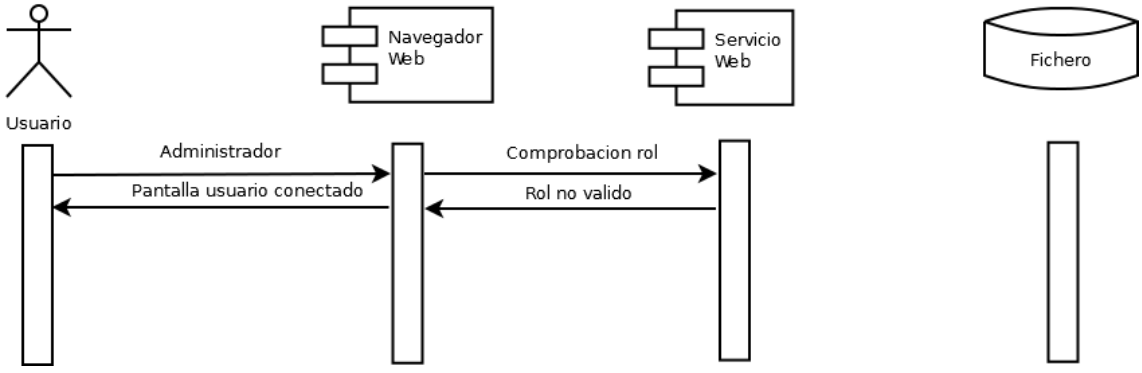

Ilustración 22: Diagrama Secuencia CU-07 (2)

CU – 08			
Título	Eliminar cualquier usuario.		
Objetivo	El administrador del sistema podrá eliminar cualquiera de los empleados.		
Pre-condiciones	• Conexión al servidor. • Usuario creado en el sistema. • Usuario logado. • Usuario con rol administrador.		
Post-condiciones	El sistema mostrará la información del usuario a consultar.		
Escenario	**1**	Ir a la pantalla inicial tras el login	
	2	Pulsar el botón administrador.	
	3	Pulsar el botón eliminar usuarios.	
	4	Introducir en el formulario el campo: • Usuario.	
	5	Pulsar en eliminar.	
Versión	1	**Fecha**	10/06/2018
Actor	Administrador		

Tabla 50: Caso de uso CU-08

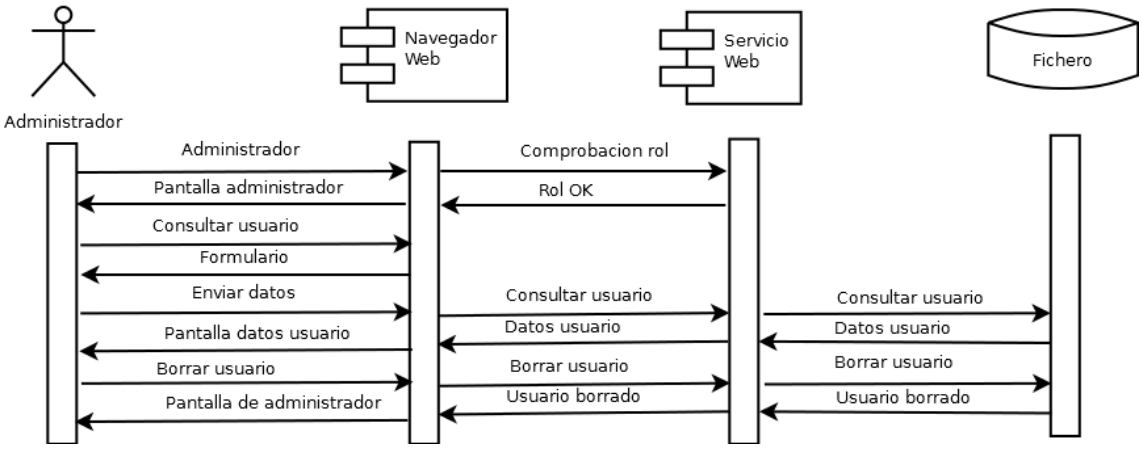

Ilustración 23: Diagrama Secuencia CU-08 (1)

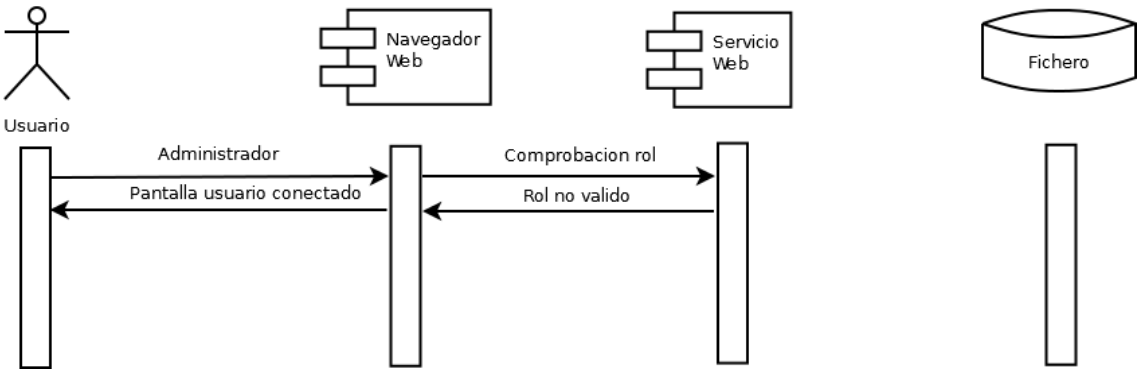

Ilustración 24: Diagrama Secuencia CU-08 (2)

5.3 Diseño físico de datos

En este apartado, vamos a definir la manera en la que los datos se almacenaran en el sistema. Es decir, en nuestro modelo-vista-controlador, formaría parte del modelo. Teniendo en cuenta los requisitos y el objetivo de nuestro trabajo, la solución escogida ha sido la siguiente.

Cómo el objetivo es que sea de fácil implantación en cualquier sistema que se vaya a desarrollar y en los requisitos lo único que se pide es almacenar datos del usuario, se determina que es insuficiente para crear una base de datos con una única tabla. Por lo tanto, se decide que el almacenamiento de datos se establecerá en un fichero, que además, nos permite implantarlo de manera más sencilla en otros sistemas sin que tenga que modificar la base de datos.

La solución de la base de datos sería más elegante, pero también consumiría un mayor número de recursos para tener una única tabla, que es lo equivalente a un fichero. En caso de ampliar este módulo con más funcionalidades en el futuro, si es necesario manejar un volumen de datos mayor, será interesante comprobar si es beneficioso para el módulo, pero en el estado actual del sistema, el fichero suple correctamente y con beneficios a la base de datos.

El fichero utilizado, esta proporcionado por la OWASP que es la desarrolladora de la API y además contiene un conjunto de métodos para poder realizar las diferentes funciones.

Los datos que se almacenan en el fichero son los siguientes:

- Id de usuario: Identificador del usuario.
- Cuenta de usuario: Nombre del usuario.
- Contraseña (Con función resumen): Contraseña almacenada con una función resumen mediante SHA-512.
- Roles: Roles del usuario.
- Si el usuario está bloqueado: Identifica si la cuenta del usuario está bloqueada o no.
- Sí el usuario está habilitado: Identifica si el usuario está habilitado y se puede utilizar.
- Contraseñas antiguas (Con función resumen): Contraseña antigua almacenada con una función resumen mediante SHA-512.

- Roles: Roles del usuario.
- Último Login: Fecha y hora del último acceso realizado correctamente.
- Último Login fallado: Fecha y hora del último acceso fallido.
- Tiempo de expiración: Tiempo para que expire la cuenta.
- Intentos de Login fallidos: Intentos de login fallidos desde su última conexión con éxito.

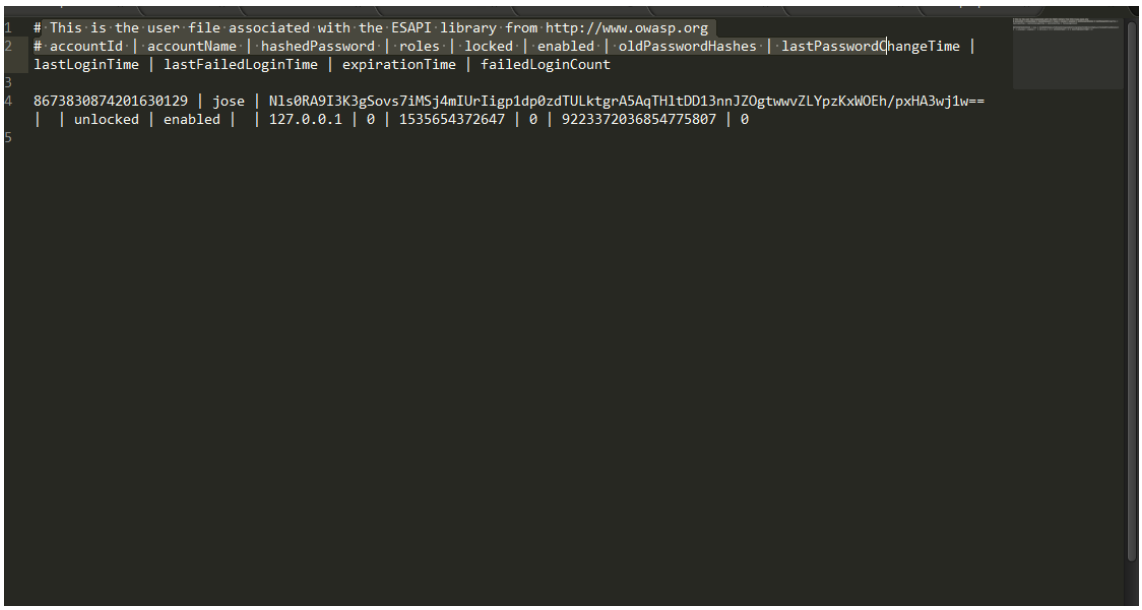

Ilustración 25: Fichero de almacenamiento de usuarios

5.4 Diseño de la interfaz.

A continuación, vamos a mostrar la interfaz básica que proporciona el sistema. Cómo nuestro sistema es un MVC, la interfaz corresponde a la parte de la vista y está formada por los distintos JSP que permiten el avance a través de la WEB y realizar toda la funcionalidad definida.

La interfaz es sencilla y lo más básica posible, ya que probablemente, al integrarlo en otro proyecto, la persona que lo diseñe, modificara el estilo de las páginas adaptándolo a su sistema. Por lo tanto, lo único que deberían hacer es incluir sus estilos en el JSP sin tocar la parte de funcionalidad que se realiza en ellos ni los identificadores.

A continuación, mostramos las diferentes pantallas y explicamos la funcionalidad existente en cada una de ellas.

5.4.1 Pantalla de Login

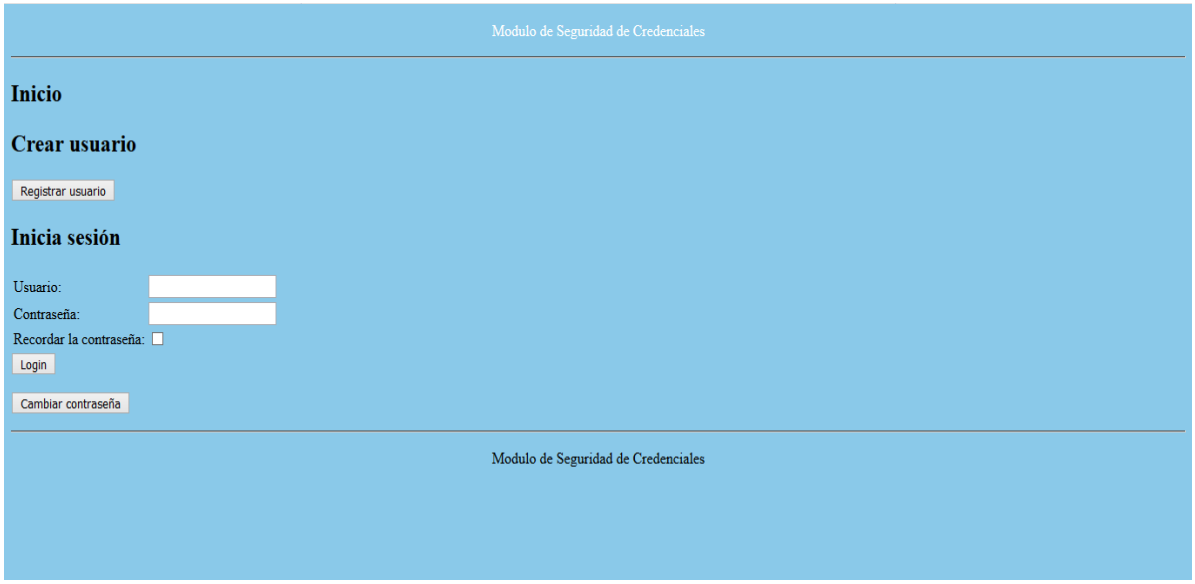

Ilustración 26: Pantalla de login

La pantalla de login será la que se muestre inicialmente al acceder a la página web. En ella se diferencian tres funcionalidades básicas.

- Crear usuario: Pulsando en el botón de Registrar Usuario, nos permitirá ir a la página de creación de usuarios que explicaremos a continuación.
- Iniciar sesión: Rellenando los campos usuario y contraseña, permitirá iniciar sesión en el sistema si los datos son correctos tras pulsar el botón de Login.
- Cambiar contraseña: Pulsando el botón nos permitirá ir a la página de modificación de contraseña que definiremos a continuación.

5.4.2 Creación de Usuario

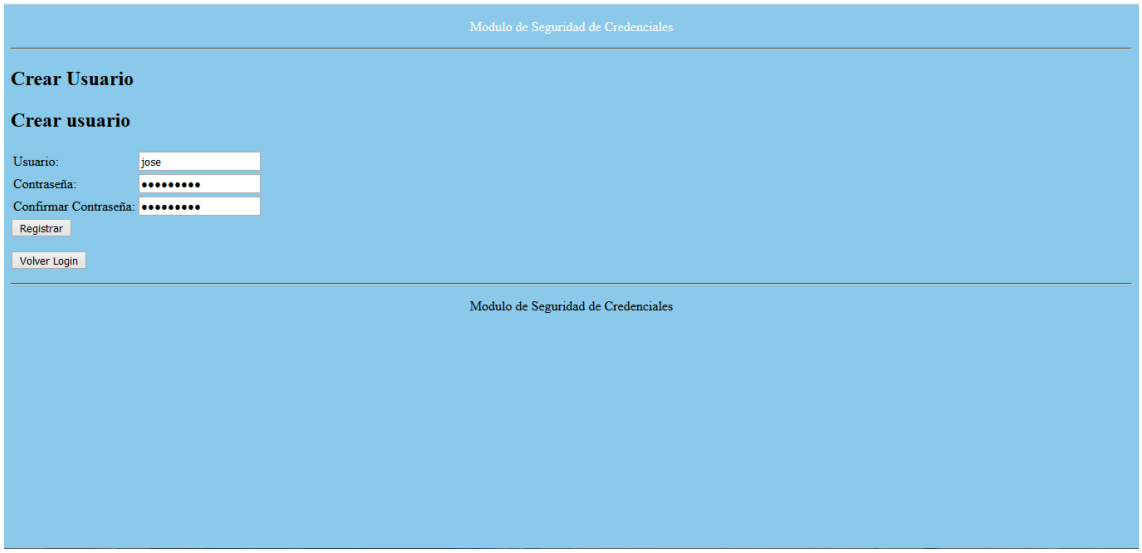

Ilustración 27: Pantalla de creación de usuario

En esta pantalla se permite crear nuevos usuarios para el sistema. Por lo tanto, las dos funcionalidades que presenta son:

- Crear usuario: Rellenando el formulario y pulsando en registrar, permite crear usuarios siempre y cuando cumplan los requisitos impuestos por el sistema.
- Volver Login: Permite volver a la pantalla de Login.

5.4.3 Cambiar contraseña

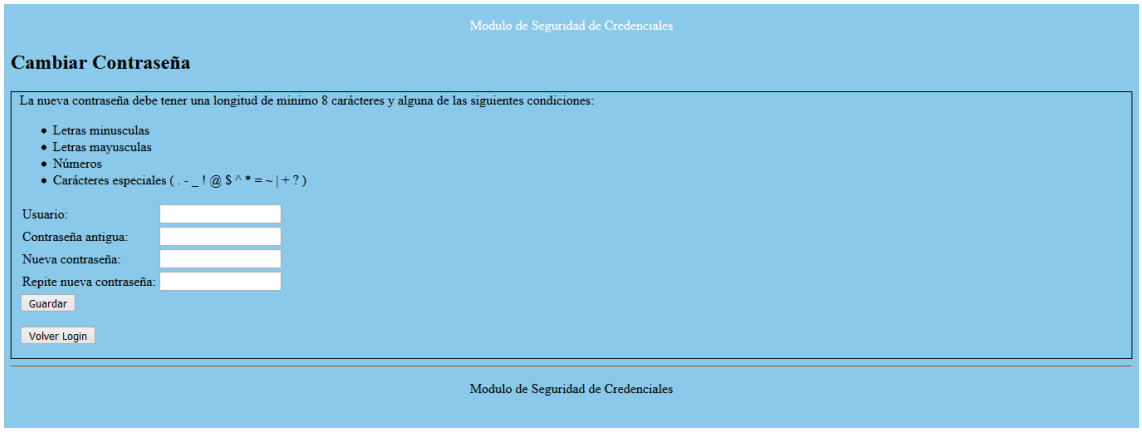

Ilustración 28: Pantalla de cambio de contraseña

En esta pantalla se permite la modificación de contraseña. Al igual que en el caso anterior, tiene dos funcionalidades.

- Cambiar contraseña: Rellenando el formulario y cumpliendo los requisitos establecidos, permitirá el cambio de contraseña al pulsar en el botón de guardar.
- Volver Login: Permite volver a la pantalla de Login.

5.4.4 Usuario conectado

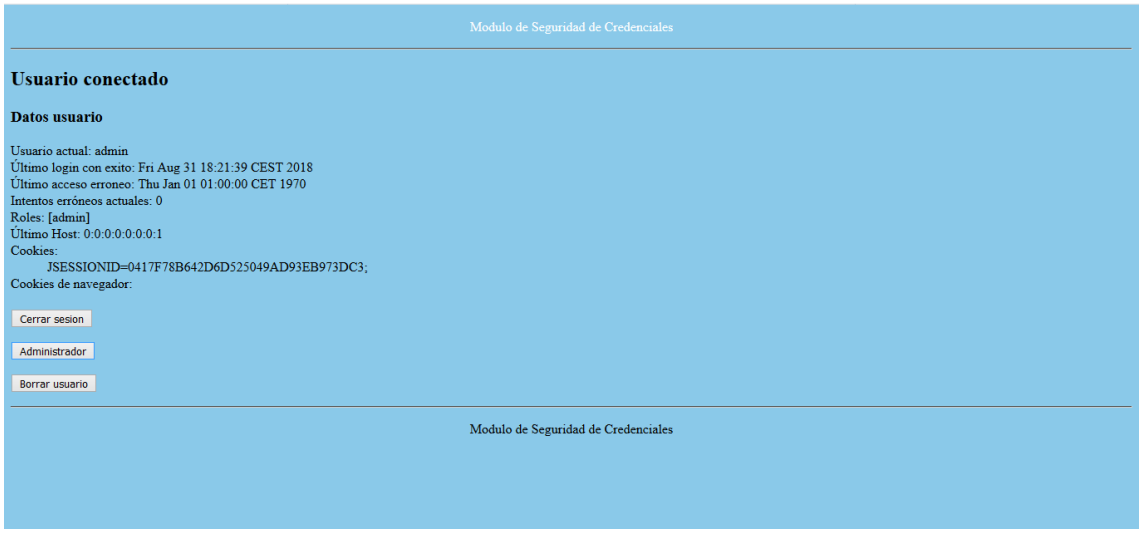

Ilustración 29: Pantalla usuario conectado

Una vez se realiza el inicio de sesión correctamente, se llevara a esta pantalla. Esta pantalla, tiene dos versiones, dependiendo de si viene desde la pantalla de Login, que mostrará la información del usuario, o sí ya estaba conectado, que no mostrará los datos del usuario.

Además, en esta página, se pueden realizar las siguientes funcionalidades:

- Cerrar sesión: Permite al usuario cerrar la sesión al pulsar sobre el botón. Enviará al usuario a la pantalla de Login.
- Administrador: Permite al usuario, en caso de tener rol de administrador, acceder a las distintas funciones que el administrador puede realizar en el sistema. Si no es usuario administrador, el botón no tendrá efecto para el usuario.
- Borrar usuario: Permite al usuario borrar su usuario si así lo considera oportuno. Se borrara su usuario del sistema permanentemente y se le enviará a la pantalla de Login de nuevo.

5.4.5 Pantalla de administrador

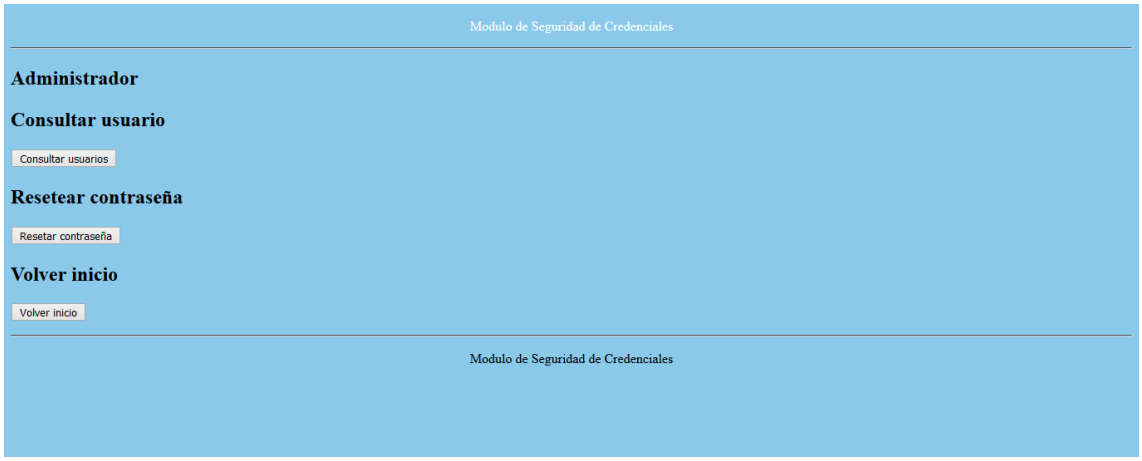

Ilustración 30: Pantalla de administrador

Una vez se accede al sistema con un usuario con rol administrador, se puede acceder a esta pantalla, que permite realizar las funciones de administrador. Las funcionalidades de esta pantalla son las siguientes:

- Consultar usuario: Permite consultar la información de un usuario del sistema en el caso que este exista. Al pulsar en el botón le llevara a la página de consulta que definiremos a continuación.
- Resetear contraseña: Permite al administrador resetear la contraseña a una que determine él, en el caso que un usuario le pida recuperar la contraseña. Al pulsar en resetear contraseña, se le enviará a una página que le permita esta modificación.

5.4.6 Usuario a consultar

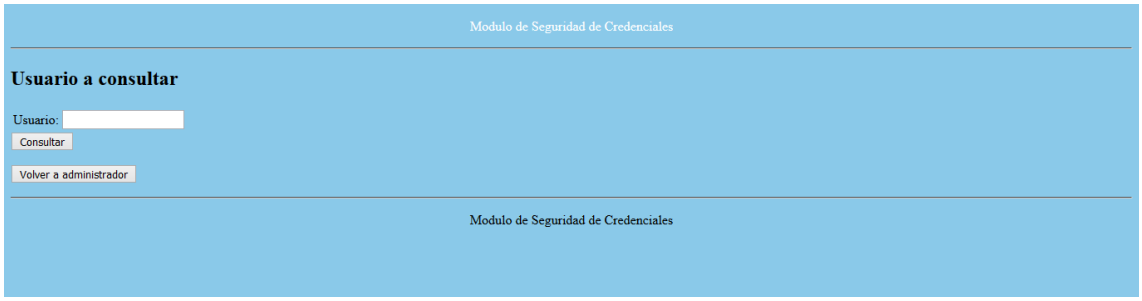

Ilustración 31: Pantalla de usuario a consultar

Esta pantalla, permitirá consultar a un usuario introduciendo su nombre de usuario en un formulario. Al pulsar sobre consultar, te enviara a la pantalla de datos de usuario. También se permite ir hacia atrás con el botón de volver a administrador.

5.4.7 Datos de usuario

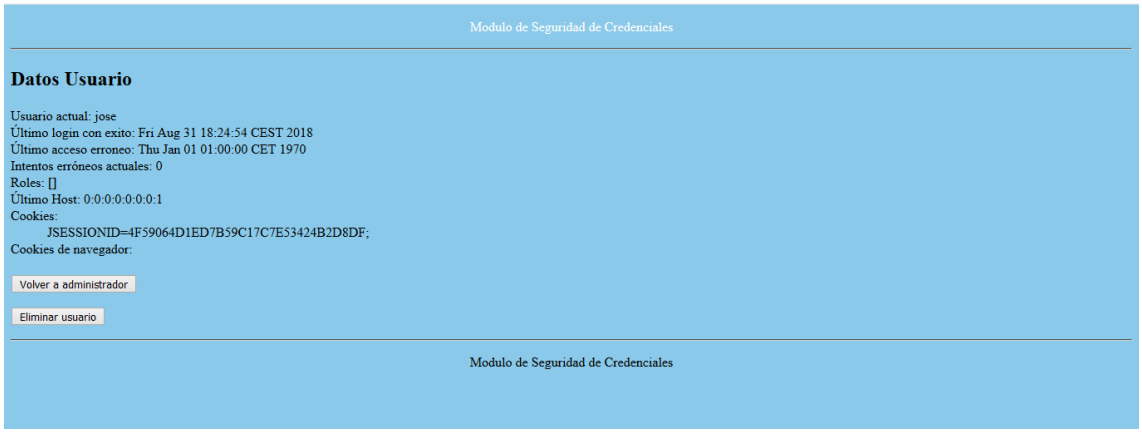

Ilustración 32: Pantalla datos de usuario

En esta pantalla se mostrará la información del usuario consultado en el caso que este exista. Esta información queda definida en los requisitos y es la que se muestra en la parte superior de la pantalla.

Las funcionalidades además de los datos mostrados en este apartado son las siguientes:

- Volver a administrador: Permite volver a la pantalla de administrador para poder realizar cualquiera de las otras funciones.
- Eliminar usuario: Permite eliminar el usuario que se ha consultado del sistema.

5.4.8 Resetear contraseña

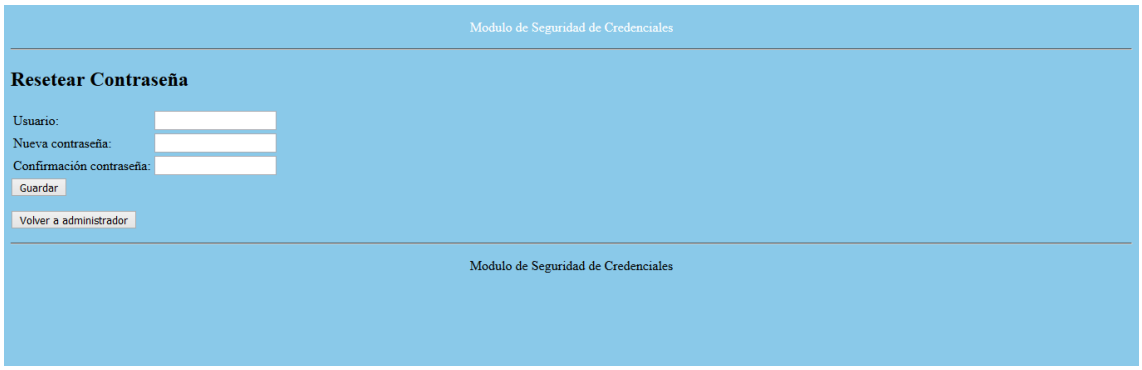

Ilustración 33: Pantalla resetear contraseña

La pantalla permite al usuario administrador resetear las contraseñas a los usuarios de la aplicación. Las funcionalidades en esta pantalla serán:

- Resetear contraseña: Permite otorgar una nueva contraseña al usuario si este la ha perdido. Para ello, deberá rellenar el formulario y pulsar en guardar. La nueva contraseña debe cumplir los requisitos de seguridad del sistema.
- Volver a administrador: Permite volver a las funcionalidades del administrador de nuevo.

6 Implementación

En este apartado se describirán los aspectos relacionados con el desarrollo del código de nuestro sistema y la posterior implantación en otros sistemas.

6.1 Descripción del código

Para la realización del código, tal y como hemos mencionado en otros apartados, utilizamos la librería ESAPI, que permite proteger la autenticación de usuarios de manera sencilla para cualquier sistema desarrollado.

Cómo el desarrollo del sistema fue nuestra primera toma de contacto con la librería (además de la documentación), se utiliza el ESAPI Swingset como base al desarrollo del sistema. El ESAPI Swingset es un pequeño tutorial que contiene laboratorios que te permite ir conociendo los métodos y te proporciona una implementación de la librería adaptándose a un fichero. Sin embargo, este tutorial ha sido modificado completamente para satisfacer las necesidades de nuestro sistema. Se ha mantenido la estructura del proyecto y la mayor implementación de métodos de la ESAPI, aunque se han tenido que modificar algunos métodos y otros añadirlos nuevos completamente.

El proyecto se divide en tres paquetes. El paquete reference, que proporciona una clase para mostrar logs del sistema (Log4LogFactory2.java), una clase que contiene la funcionalidad de los usuarios, que ha debido ser modificada para el correcto funcionamiento del host (DefaultUser.java) y una clase que implementa métodos de la librería adaptados a un fichero (FileBasedAuthenticator2.java). El paquete setup, que contiene un setup.java permite arrancar el sistema correctamente. Por último, el paquete tfg, que tendrá la clase controller.java que será el servlet de nuestro sistema. Además, existen múltiples JSP en los que se incluye tanto código con funcionalidad, cómo la interfaz de las páginas.

El paquete de setup y reference se mantiene igual que en el tutorial a excepción de la clase DefaultUser.java mencionada anteriormente y FileBasedAuthenticator2.java, en la que además de los métodos que ya nos proporcionaba (login, obtener usuario, registrar, borrar usuario…) se han tenido que añadir métodos nuevos, cómo el reseteo de contraseña para que el administrador pueda realizar los cambios, la verificación de que la contraseña de cambio es segura y se ha modificado el método de cambio de contraseña permitiendo almacenar los cambios en el fichero.

En cuanto al controller.java, que como mencionamos antes, es el servlet de nuestro sistema, es completamente nuevo. En él, se realizan las distintas funciones del sistema a través de identificadores ocultos que se obtienen en el JSP. Así permite realizar las distintas funcionalidades y llamar correctamente a los JSP para realizar de manera adecuada las funcionalidades gracias al código existente en el JSP.

Por último, los JSP como mencionamos anteriormente estarán relacionados con las funcionalidades, es decir, cada JSP es para una funcionalidad distinta del sistema. Por ejemplo, existe el JSP de creación de usuarios que permite recoger los datos de creación y mediante el servlet y la funcionalidad añadida en el JSP, registrar al usuario correctamente en el sistema.

Como se menciona en el apartado de diseño de la interfaz, para modificar está, deberán modificar los JSP existentes. A continuación, mostramos una lista con los JSP, que tienen un nombre descriptivo.

- Admin.jsp: Tiene la funcionalidad de la pantalla de administrador.
- CambiarContraseña.jsp: Tiene la funcionalidad de la pantalla de cambiar contraseña.
- ConsultarUsuario.jsp: Tiene la funcionalidad de la pantalla de consultar datos del administrador.
- CreateUser.jsp: Tiene la funcionalidad de la pantalla de crear usuario.
- DatosUsuario.jsp: Muestra los datos del usuario consultado por el administrador.
- footer.jsp: Contiene el estilo del footer.
- Inicio.jsp: Tiene la funcionalidad de la pantalla de inicio, lo que correspondería al login.
- LoginAceptado: Tiene la funcionalidad de la pantalla posterior a realizar el login.
- ResetContraseña: Tiene la funcionalidad de resetear la contraseña por parte del administrador.

Además de lo mencionado anteriormente, hay que tener en cuenta el fichero en el que se van a registrar los usuarios y el que indica las url a las que puede acceder el administrador.

Por otro lado tambien hay que tener en cuenta los ficheros de propiedades de la librería y del servidor. Estos ficheros y resto de carpetas que hay que añadir se explicarán en el siguiente punto de implantación del sistema.

6.2 Implantación del sistema

Para la implantación del sistema se realizara un fichero ZIP con todo lo necesario para que el sistema funcione correctamente. Este ZIP estará formado por el proyecto que habrá que importar en eclipse, un almacén de claves, una carpeta con la configuración de ESAPI y por último, una carpeta con el servidor de Tomcat, ya que tiene ciertas modificaciones.

El primer paso para la implantación del sistema es copiar el almacén de claves (.keystore) y la carpeta de configuración de ESAPI (.esapi) en la ruta C:\Usuarios\[Nombre_Usuario]. A continuación, se muestra unas rutas de ejemplo.

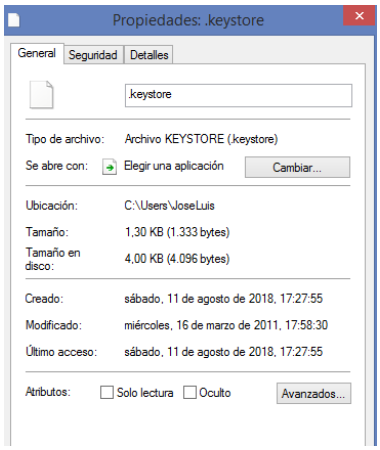

Ilustración 34: Propiedades .esapi

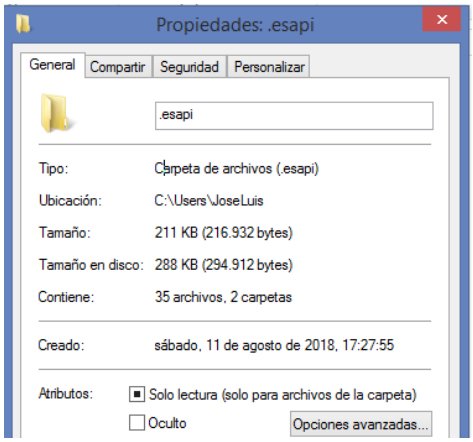

Ilustración 35: Propiedades .keystore

Dentro de la carpeta de configuración de ESAPI se encuentran varios ficheros y carpetas. A continuación se mencionan los más importantes y los que pueden necesitar modificarse dependiendo del sistema en el que se incluya.

- ESAPI.properties: Contiene las propiedades de la librería. Por ejemplo el método de cifrado, el método de HASH, numero de fallos al iniciar sesión… En caso de que un sistema concreto necesite un tipo de cifrado distinto, se podría modificar por ejemplo.
- validation.properties: Contiene las validaciones del sistema. Es decir, expresiones regulares que determinen el formato de ciertos datos de entrada.
- users.txt: Es el fichero en el que se almacenan los usuarios y su información según se van registrando en el sistema.
- Carpeta fbac-polices: Contiene distintos ficheros que permiten el bloqueo a información a través de roles. Por ejemplo, el fichero URLAccessRules permite o bloquea el acceso a ciertas URL dependiendo del usuario. Por ejemplo, nuestro

sistema la utiliza para que solo el administrador pueda acceder a la pantalla de administrador.

Cómo hemos mencionado anteriormente, en la carpeta de configuración de ESAPI hay más carpetas y ficheros, pero los mencionados son los que serán útiles en caso de modificarse en nuestro sistema.

Tras esto, se puede importar el proyecto en eclipse. Para ello se importará la carpeta TFG en eclipse como proyecto existente. A continuación se muestran capturas de cómo se realizaría.

Pulsamos en importar y seleccionamos Existing Projects into Workspace.

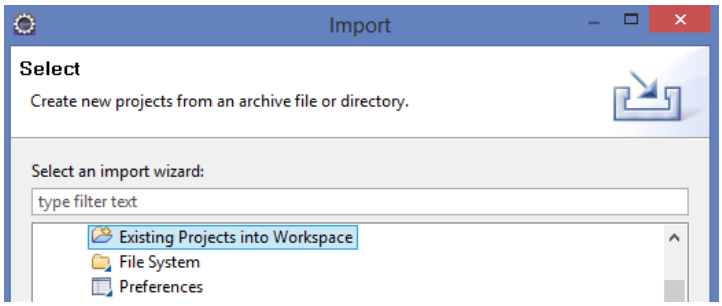

Ilustración 36: Importar proyecto

Tras esto, buscamos el directorio raíz, que estará dentro de la carpeta tfg y tendrá el mismo nombre, TFG, cómo podemos ver en la captura. Una vez seleccionemos la carpeta y pulsemos finish, el proyecto se incluirá en eclipse.

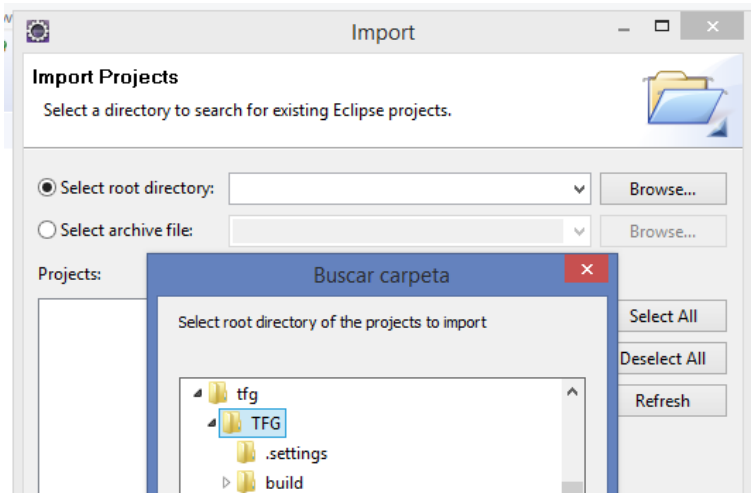

Ilustración 37: Seleccionar raíz de proyecto

Una vez se ha importado correctamente el proyecto en eclipse, se podrá modificar el código en el caso que el sistema a implementar lo requiera.

Por último, faltaría añadir el servidor. En nuestro caso, proporcionamos apache-Tomcat 9.0.10, ya que el proyecto tiene sus propias configuraciones del servidor. En caso de querer utilizar otra versión de Tomcat no habría problema siempre y cuando se modifiquen los ficheros de configuración de manera que queden similares a los proporcionados. En caso de utilizar otros servidores, por ejemplo Glassfish, habría que

adaptar al igual que en Tomcat las configuraciones del servidor, para que permita su correcto funcionamiento

Para añadir el servidor en eclipse, habrá que moverse en la parte inferior a la pantalla de servers y pulsar con el botón derecho para crear uno nuevo.

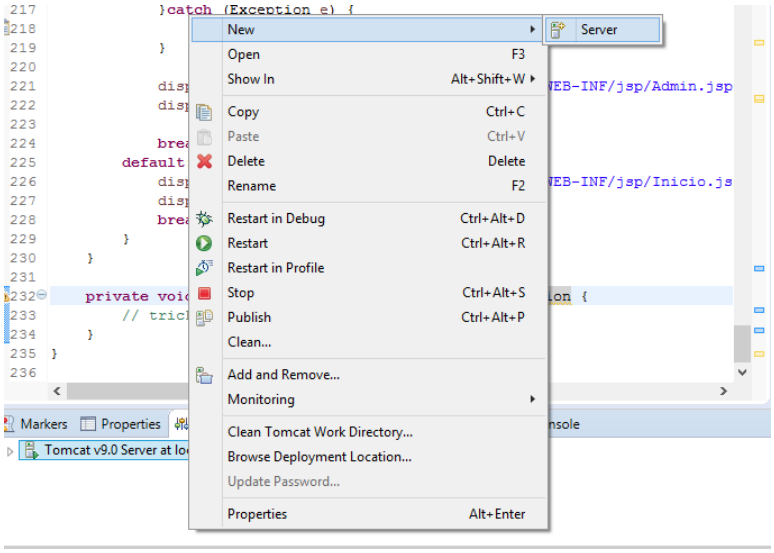

Ilustración 38: Añadir servidor

Una vez pulsado el nuevo servidor, elegimos el tipo, en este caso Tomcat v9.0.

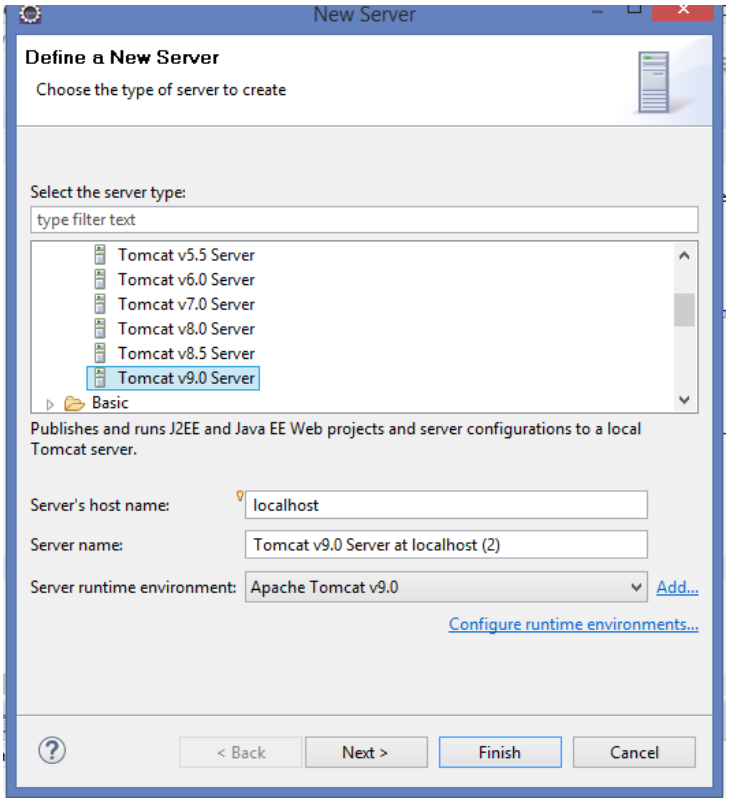

Ilustración 39: Seleccionar servidor

En caso de no venir por defecto la ruta en la que tenemos Tomcat, habrá que pulsar Add e introducir la ruta en la que lo tenemos cómo vemos en la imagen inferior. Tras esto, pulsamos en finalizar.

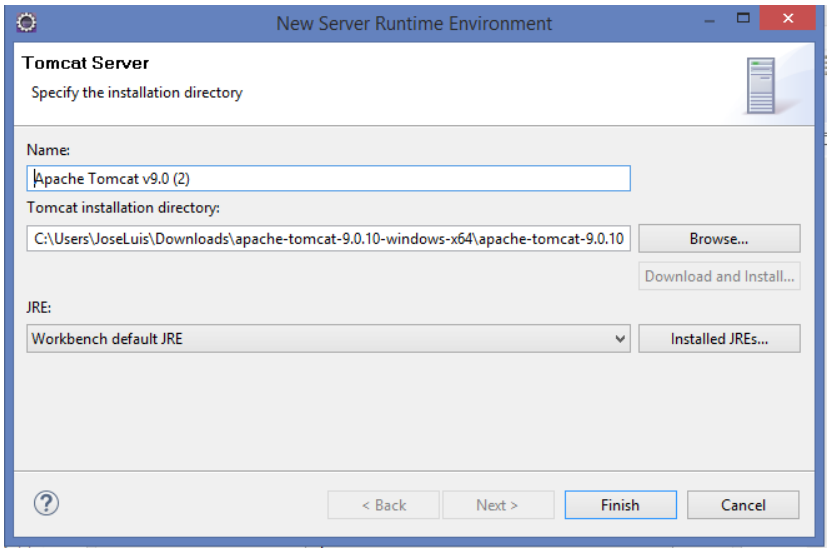

Ilustración 40: Ruta del servidor

Por ultimo añadimos el proyecto a la parte de Configured, y ya estará disponible para lanzarse a través del servidor. Por último damos a finalizar.

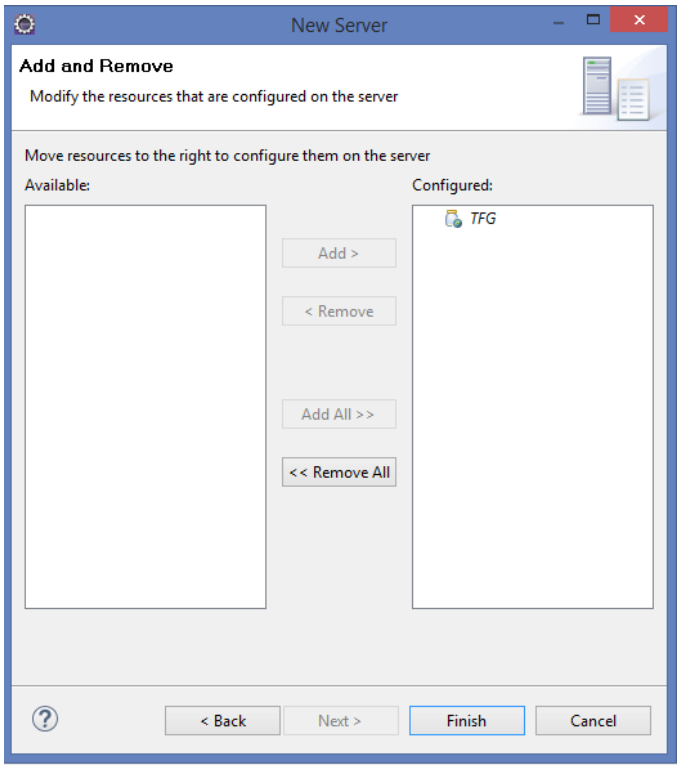

Ilustración 41: Añadir proyecto al servidor

Con esto el sistema estaría completamente operativo en cualquier entorno y se podría introducir en cualquier sistema. En el caso de necesitar modificaciones en el

sistema, se podrían realizar desde eclipse o en cualquiera de los ficheros mencionados anteriormente.

7 Evaluación

Para la correcta evaluación del sistema, se realizaran un conjunto de casos de prueba de pruebas funcionales para verificar así, que todos los requisitos del sistema se cumplen de manera satisfactoria. A continuación, mostramos una plantilla que vamos a utilizar para estos casos de prueba y posteriormente realizaremos una matriz de trazabilidad para verificar que se han probado todos los requisitos.

7.1 Casos de prueba

CP-XX	
Objetivo	
Entrada	
Secuencia de pasos	
Salida	
Resultado	
Requisitos relacionados	

Tabla 51: Plantilla de casos de prueba

La tabla de casos de prueba está formada por los siguientes elementos:

- **Identificador:** Identificará cada caso de pruebas de manera univoca. Se localiza en la parte superior de la tabla. Su formato será CP-XX, siendo XX un número de dos cifras incremental desde el valor 01.
- **Objetivo:** Define la acción que se va a probar.
- **Entrada:** Define los datos que se introducen para realizar la prueba.
- **Secuencia de pasos:** Define los pasos que se realizan para realizar la prueba.
- **Salida:** Contiene los cambios realizados tras la prueba.
- **Resultado:** OK o KO dependiendo de si la prueba pasa de manera exitosa o no.
- **Requisitos relacionados:** Requisitos software con los que va enlazado las pruebas.

CP-01	
Objetivo	Crear usuario correctamente
Entrada	Se introducen datos correctos de usuario.
Secuencia de pasos	1. Pulsar en crear usuario 2. Introducir los datos del usuario 3. Pulsar en registrar
Salida	El usuario se ha registrado en el fichero y permite acceder al sistema.
Resultado	OK
Requisitos relacionados	RS-F001,RS-F003, RS-NF002, RS-NF008, RS-NF009, RS-NF010, RS-NF011 y RS-NF012

Tabla 52: Caso de prueba CP-01

CP-02	
Objetivo	Crear usuario incorrectamente
Entrada	Se introducen datos de usuario con una contraseña sin seguridad.
Secuencia de pasos	1. Pulsar en crear usuario 2. Introducir los datos del usuario 3. Pulsar en registrar
Salida	El usuario no se registra en el sistema ya que no cumple las restricciones de contraseña
Resultado	OK
Requisitos relacionados	RS-F003, RS-NF001 RS-NF008, RS-NF009, RS-NF010 y RS-NF012

Tabla 53: Caso de prueba CP-02

CP-03	
Objetivo	Modificar contraseña correctamente
Entrada	Se introducen los datos de usuario y una contraseña que cumple los requisitos.
Secuencia de pasos	1. Pulsar en cambiar contraseña 2. Introducir los datos del usuario 3. Pulsar en guardar
Salida	El sistema modificará el usuario y actualizara la nueva contraseña.
Resultado	OK
Requisitos relacionados	RS-F002 , RS-F004, RS-NF002, RS-NF008, RS-NF009, RS-NF010, RS-NF011 y RS-NF012

Tabla 54: Caso de prueba CP-03

CP-04	
Objetivo	Modificar contraseña incorrectamente
Entrada	Se introducen los datos de usuario y una contraseña que no cumple los requisitos.
Secuencia de pasos	1. Pulsar en cambiar contraseña 2. Introducir los datos del usuario 3. Pulsar en guardar
Salida	El sistema no modificara el usuario.
Resultado	OK
Requisitos relacionados	RS-F004, RS-NF004, RS-NF008, RS-NF009, RS-NF010 y RS-NF012

Tabla 55: Caso de prueba CP-04

CP-05	
Objetivo	Iniciar sesión correctamente
Entrada	Se introducen los datos de un usuario registrado en el sistema.
Secuencia de pasos	1. Introducir los datos del usuario 2. Pulsar en login.
Salida	El sistema iniciara sesión y modificara el fichero de usuarios.
Resultado	OK
Requisitos relacionados	RS-F002, RS-F005, RS-NF008, RS-NF009, RS-NF010, RS-NF011 y RS-NF012

Tabla 56: Caso de prueba CP-05

CP-06	
Objetivo	Iniciar sesión incorrectamente
Entrada	Se introducen los datos de un usuario registrado en el sistema con contraseña errónea.
Secuencia de pasos	1. Introducir los datos del usuario 2. Pulsar en login. 3. Repetir pasos 1 y 2 al menos cuatro veces.
Salida	El sistema bloqueará el usuario y modificara el fichero de usuarios.
Resultado	OK
Requisitos relacionados	RS-F002, RS-F005, RS-NF003, RS-NF008, RS-NF009, RS-NF010, RS-NF011 y RS-NF012

Tabla 57: Caso de prueba CP-06

CP-07	
Objetivo	Borrar Usuario
Entrada	Borrar el usuario con sesión iniciada.
Secuencia de pasos	1. Iniciar sesión con el usuario. 2. Una vez iniciada la sesión pulsar en borrar usuario.
Salida	El sistema eliminara el usuario y será eliminado del fichero.
Resultado	OK
Requisitos relacionados	RS-F002, RS-F007, RS-NF003, RS-NF008, RS-NF009, RS-NF010, RS-NF011 y RS-NF012

Tabla 58: Caso de prueba CP-07

CP-08	
Objetivo	Cerrar Sesión
Entrada	Usuario con sesión iniciada.
Secuencia de pasos	1. Iniciar sesión con el usuario. 2. Una vez iniciada la sesión pulsar en el botón cerrar sesión.
Salida	El sistema cerrará sesión y será modificado en el fichero de usuarios.
Resultado	OK
Requisitos relacionados	RS-F002, RS-F006, RS-NF003, RS-NF008, RS-NF009, RS-NF010, RS-NF011 y RS-NF012

Tabla 59: Caso de prueba CP-08

CP-09	
Objetivo	Consultar usuario
Entrada	Usuario administrador con sesión iniciada
Secuencia de pasos	1. Pulsar en administrador. 2. Pulsar en consultar usuario. 3. Introducir los datos del usuario a consultar. 4. Recuperar los datos.
Salida	El sistema consultará los datos del usuario.
Resultado	OK
Requisitos relacionados	RS-F008, RS-F009, RS-NF007, RS-NF008, RS-NF009, RS-NF010, RS-NF011 y RS-NF012

Tabla 60: Caso de prueba CP-09

CP-10	
Objetivo	Consultar usuario (No administrador)
Entrada	Usuario no administrador con sesión iniciada
Secuencia de pasos	1. Pulsar en administrador. 2. Pulsar en consultar usuario. 3. Introducir los datos del usuario a consultar. 4. Recuperar los datos.
Salida	El sistema no permitirá acceder a la funcionalidad de administrador.
Resultado	OK
Requisitos relacionados	RS-F008, RS-F009, RS-NF007, RS-NF008, RS-NF009, RS-NF010 y RS-NF012

Tabla 61: Caso de prueba CP-10

CP-11	
Objetivo	Resetear contraseña
Entrada	Usuario administrador con sesión iniciada
Secuencia de pasos	1. Pulsar en administrador. 2. Pulsar en resetear contraseña. 3. Introducir los datos para cambiar contraseña. 4. Pulsar en guardar.
Salida	El sistema actualizará los datos del usuario en el fichero.
Resultado	OK
Requisitos relacionados	RS-F002, RS-F008, RS-F010, RS-NF006, RS-NF008, RS-NF009, RS-NF010, RS-NF011 y RS-NF012

Tabla 62: Caso de prueba CP-11

CP-12	
Objetivo	Resetear contraseña (No administrador)
Entrada	Usuario no administrador con sesión iniciada
Secuencia de pasos	1. Pulsar en administrador. 2. Consultar usuario. 3. Introducir los datos del usuario a consultar. 4. Recuperar los datos.
Salida	El sistema no permitirá acceder a la funcionalidad de administrador.
Resultado	OK
Requisitos relacionados	RS-F008, RS-F010, RS-NF003, RS-NF008, RS-NF009, RS-NF010, y RS-NF012

Tabla 63: Caso de prueba CP-12

CP-13	
Objetivo	Eliminar usuario
Entrada	Usuario administrador con sesión iniciada
Secuencia de pasos	1. Pulsar en administrador. 2. Pulsar en consultar usuario. 3. Introducir los datos del usuario a consultar. 4. Eliminar usuario.
Salida	El sistema eliminará al usuario del sistema y del fichero de usuarios.
Resultado	OK
Requisitos relacionados	RS-F002, RS-F008, RS-F011, RS-NF008, RS-NF009, RS-NF010, RS-NF011 y RS-NF012

Tabla 64: Caso de prueba CP-13

CP-14	
Objetivo	Eliminar usuario (No administrador)
Entrada	Usuario no administrador con sesión iniciada
Secuencia de pasos	1. Pulsar en administrador. 2. Pulsar en consultar usuario. 3. Introducir los datos del usuario a consultar. 4. Recuperar los datos.
Salida	El sistema no permitirá acceder a la funcionalidad de administrador.
Resultado	OK
Requisitos relacionados	RS-F008, RS-F011, RS-NF003, RS-NF008, RS-NF009, RS-NF010 y RS-NF012

Tabla 65: Caso de prueba CP-14

CP-15	
Objetivo	Usuario administrador único
Entrada	Ninguna
Secuencia de pasos	1. Visualizar el fichero de usuarios para comprobar roles.
Salida	Ninguna
Resultado	OK
Requisitos relacionados	RS-NF005

Tabla 66: Caso de prueba CP-15

CP-16	
Objetivo	Implementación del sistema.
Entrada	Ninguna
Secuencia de pasos	1. Implantar el sistema en un equipo distinto al de desarrollo.
Salida	Ninguna
Resultado	OK
Requisitos relacionados	RS-NF013

Tabla 67: Caso de prueba CP-16

7.2 Matriz de trazabilidad RS-CP

	CP-01	CP-02	CP-03	CP-04	CP-05	CP-06	CP-07	CP-08	CP-09	CP-10	CP-11	CP-12	CP-13	CP-14	CP-15	CP-16
RS-F001	x															
RS-F002			x		x	x	x	x			x		x			
RS-F003	x	x														
RS-F004			x	x												
RS-F005					x	x										
RS-F006								x								
RS-F007							x									
RS-F008									x	x	x	x	x	x		
RS-F009									x	x						
RS-F010											x	x				
RS-F011													x	x		
RS-NF001		x														
RS-NF002	x		x													
RS-NF003						x	x	x			x	x	x	x		
RS-NF004				x												
RS-NF005															x	
RS-NF006											x					
RS-NF007									x	x						
RS-NF008	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
RS-NF009	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
RS-NF010	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
RS-NF011	x		x		x	x	x	x	x		x		x			
RS-NF012	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
RS-NF013																x

Tabla 68: Matriz trazabilidad RS-CP

8 Gestión del proyecto

8.1 Modelo de desarrollo de software y metodología

El modelo de desarrollo de software queda definido en el punto 4.1 Ciclo de Vida. Por lo tanto, no es necesario volver a incluirlo en este apartado. La metodología utilizada como guía a la hora de realizar el documento ha sido Métrica Versión 3, proporcionada por los profesores de la asignatura dirección de proyectos de desarrollo de software (DPDS) de la UC3M.

8.2 Planificación

La planificación se lleva a cabo para entregar el trabajo en la tercera convocatoria del curso 2017-2018, por lo tanto el 25 de Septiembre. Empezando el proyecto a inicios del mes de Abril. La asignatura de TFG tiene un tiempo aproximado de 300 horas.

Por lo tanto, dejando 10 días de margen final para la revisión, es decir, establecer la fecha límite el 15 de Septiembre, el proyecto tendrá una duración de 24 semanas, siendo 12 horas 30 minutos semanales.

Las tareas a realizar han sido las siguientes:

- Análisis del estado del arte.
- Elección de la solución.
- Análisis del sistema.
- Diseño del sistema.
- Implementación del sistema.
- Evaluación del sistema.

A continuación mostramos el diagrama de Gantt realizado con ganttproject. [18]

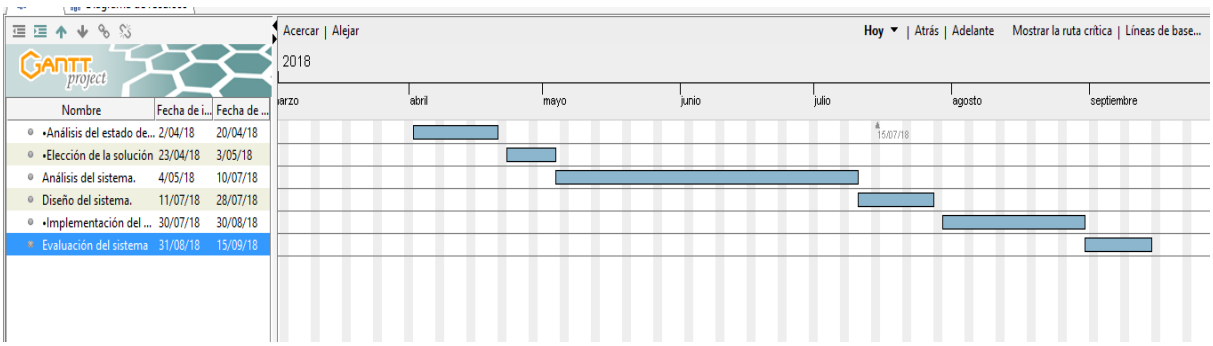

Ilustración 42: Diagrama de Gantt

8.3 Presupuesto

8.3.1 Personal del proyecto

Estimamos el coste asociado al personal a la hora de realizar el proyecto. Para ello, utilizamos un salario por hora medio de un desarrollador.

Nombre	Horas dedicadas al proyecto	Coste/Hora	Sueldo Bruto
Jose Luis Parra Olmedilla	300	9,69€	2.907,00€
Total	300	9,69€	2.907,00€

Tabla 69: Costes personal del proyecto

8.3.2 Equipo de trabajo

A continuación, estimamos el coste de los equipos necesarios para realizar el proyecto. En la siguiente tabla, mostramos los costes totales del equipo.

Producto	Descripción	Unidades	Coste/Unidad	Coste total
Ordenadores portátiles	Se utilizará para el desarrollo de documentación.	1	700,00€	700,00€
Periféricos	Se incluye alfombrillas y ratones.	1	25,00€	25,00€
Impresora	Se utilizará para la impresión de documentos.	1	80,00€	80,00€
Licencia Microsoft Office	Se utilizará para la realización de documentos	1	Licencia Universidad	0,00€

Tabla 70: Costes equipo de trabajo

Como este equipo se va a utilizar durante un mayor periodo de tiempo, el coste para el proyecto debe ser proporcional al tiempo que se va a utilizar en él. Es decir, el coste que tiene el equipo los seis meses de proyecto.

Producto	Coste total	Tiempo de amortización	Coste para el proyecto
Ordenadores portátiles	700,00€	48 meses	87,50€
Periféricos	25,00€	12 meses	12,50€
Impresora	80,00€	36 meses	13,33€
Total	805,00€		113,33€

Tabla 71: Amortización equipo de trabajo

8.3.3 Material fungible

Corresponde a los materiales que se van a utilizar durante el proyecto para la realización de documentos, tomar notas…

Producto	Descripción	Unidades	Coste total
Paquete 500 folios	Se utilizará para la impresión de documentos.	2	6,00€
Tinta	Se utilizará para la impresión de documentos.	1	64,52€
Bolígrafos	Materiales para escribir notas.	2	1,00€
Total			71,52€

Tabla 72: Costes material fungible

8.3.4 Costes indirectos

En los costes indirectos de este proyecto, únicamente añadimos los costes de la luz utilizada para enchufar los equipos, ya que al realizarse desde nuestra propia casa, el resto de gastos es el mismo.

Concepto	Descripción	Coste/mes	Nº meses	Coste total
Luz	Gasto por la luz.	30,00€	6	180,00€
Total		30,00€		180,00€

Tabla 73: Costes indirectos

8.3.5 Coste total del proyecto

Por último, mostramos un resumen con el coste total del proyecto.

Resumen de costes	
Coste total por personal	2.907,00€
Coste total del equipo	113,33€
Coste total del material fungible	71,52€
Coste total indirecto	180,00€
Coste total del proyecto	3.271,85€

Tabla 74: Coste total del proyecto

8.4 Entorno socio-económico

En este apartado vamos a analizar el impacto económico y social que puede tener nuestro proyecto en la actualidad.

En cuanto al impacto económico, cabe destacar que es un proyecto desarrollado con una librería accesible por todo el mundo y mediante código JAVA. Es decir, económicamente es un sistema sencillo ya que no tiene costes de licencias.

Por lo tanto, el proyecto ha sido costoso en términos de gastos de personal para la realización del sistema, pero una vez realizado, la adaptación de este a otros sistemas es muy inferior al tiempo en realizarlo.

En cuanto al impacto social, como se ha comentado a lo largo del trabajo, la protección de credenciales es un tema en el que se debe concienciar tanto a los usuarios de los sistemas, cómo a los desarrolladores de los sistemas.

En la actualidad estamos viviendo un incremento de uso de aplicaciones, elementos comunes de la vida pasan a tener conectividad a la red (IOT), ciudades ubicuas, etc. Esto provocará que perder una credencial de usuario sea comparable a perder un DNI en la actualidad por ejemplo.

El sistema proporcionado está orientado al uso de los TFGs para tratar de fomentar la conciencia de proteger todos los datos sensibles del usuario. Este sistema también se podría utilizar para pequeños proyectos para facilitar el manejo de datos sensibles.

Por lo tanto, podemos concluir que tanto en el entorno económico como en el social, el impacto de nuestro sistema es positivo.

8.5 Marco legal

A continuación mostramos las leyes que afectan al funcionamiento del sistema y se deben tener en cuenta.

- **Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal** [19]**:** La ley obliga a proteger los datos personales sensibles de nuestro sistema. En nuestro caso, la contraseña del usuario es un dato sensible. Para ello nuestro sistema utiliza un cifrado AES y se almacenarán los datos en el fichero mediante funciones de resumen SHA-512.

9 Conclusiones

Para finalizar el trabajo, exponemos los aspectos que nos ha proporcionado el trabajo, retos a los que nos hemos enfrentado y dificultades que hemos tenido. Además, añadiremos un apartado con futuras mejoras que puede tener el sistema para el que hemos trabajado.

El trabajo ha permitido extraer un nuevo conocimiento en referencia a la autenticación y la protección de credenciales. Tras la realización del capítulo del estado del arte, observe como ha habido una evolución constante para el tratamiento de esta información. Además, como podemos observar, con el paso del tiempo, cada vez se utilizan sistemas que te proporcionan mecanismos de seguridad, como los IAM. Por lo tanto, se ha facilitado mucho el trabajo de desarrolladores de software, ya que pueden integrar estas herramientas en su sistema.

En cuanto a los objetivos que nos proponemos en el apartado de la introducción, se han cumplido de manera satisfactoria. Se ha conseguido que el sistema sea lo más sencillo posible, permitiendo las funcionalidades básicas para cualquier sistema. La interfaz es muy sencilla para que el usuario pueda moverse entre pantallas de manera intuitiva. Con esto, intentamos fomentar el uso de este tipo de herramientas en futuros TFGs.

El mayor problema del TFG fue entender el funcionamiento de la librería utilizada. ESAPI es una librería fácil de utilizar, pero es difícil de comprender su funcionamiento inicialmente. La mejor manera de entenderla fue mediante un tutorial que proporcionan los creadores de esta librería. Una vez consigues instalar el tutorial, te permite realizar actividades para realizar distintas tareas, además de proporcionar una implementación más entendible de la librería. Por lo tanto, gracias a este tutorial, se puede desarrollar de una manera sencilla un sistema.

Sin embargo, la conclusión de este tipo de trabajo siempre es positiva. Tener que enfrentarte a un problema nuevo es lo que realmente nos aporta el conocimiento. Por ejemplo, cómo hemos indicado antes, en el trabajo del arte nos ha enseñado diferentes herramientas, la capacidad de buscar soluciones a problemas, como el caso de encontrar un tutorial para entender la librería. Además nos aporta una idea de lo que es un desarrollo de un pequeño proyecto.

9.1 Trabajos futuros

Una vez concluido el trabajo, existen elementos que pueden ser mejorados, nuevas funcionalidades que no se ajustan a lo que buscábamos, o elementos para los que se necesita más tiempo. A continuación, enumeramos algunos de estos elementos.

- Añadir funcionalidad para trabajar con roles: Esta funcionalidad no se añadió porque haría más complejo el sistema. Pero en el caso de querer ampliar el sistema, podría ser una nueva característica importante para poder diferenciar capacidades entre los distintos usuarios.
- Introducir una base de datos: En el caso de nuestro sistema, cómo los datos a almacenar son muy pocos y únicamente sería una tabla, la base de datos penalizaría más el sistema que mejorarlo. En el caso de añadir una funcionalidad de roles como mencionamos en el punto anterior, que habría

que almacenar más datos, cobraría más sentido tener que introducir datos en la base de datos.

- Mejorar la interfaz: En nuestro sistema, cómo se va a integrar en distintos proyectos, la interfaz será modificada para ponerla similar al del proyecto en el que se integra. Sin embargo, se podría intentar generar una interfaz más genérica que se adaptase a cualquier sistema.
- Publicar el repositorio del proyecto: El día de la exposición frente al tribunal, se publicara el proyecto que actualmente se encuentra en modo privado en la siguiente URL: https://bitbucket.org/Josepaol/tfg-jose-luis/src/master/.

ANEXO I: Acrónimos

PAP: Password Authentication Protocol.

MFA: Multi-Factor Authentication.

IAM: Identity and access management.

IOT: Internet de las cosas.

SSO: Single Sign On.

URL: Localizador Uniforme de Recursos (Uniform Resource Locator).

XRI: eXtensible Resource Identifier.

XML: Extensible Markup Language.

SAML: Security Assertion Markup Language.

DPDS: Dirección de proyectos de desarrollo de software.

UC3M: Universidad Carlos III de Madrid.

JSP: JavaServers Pages.

ANEXO II: Colección de TFG analizados

A continuación mostramos una tabla con los TFG analizados y sus valoraciones:

Título	Autor	Año	Valoración
Optimización de consultas en bases de datos relacionales [20]	García Frutos, Raquel	2016	Mencionan la ley, pero no indican el cómo la aplican
Desarrollo de gestor de comidas a domicilio mediante la aplicación de mensajería instantánea Telegram [21]	Rivero Ortiz, Antonio Pablo	2016	Mencionan la ley, pero no indican el cómo la aplican
Modelado de un puerto marítimo con Unity 3D [22]	Bernárdez Martínez, Alejandro	2016	Mencionan la ley, pero no indican el cómo la aplican
Continuous authentication based on data from smart devices [23]	Barbero Rodríguez, Silvia	2017	Protegen credenciales de usuario
Aplicación nutricional y seguimiento deportivo, en atletas de alto rendimiento [24]	Valdes Fernández, José	2016	No protegen las credenciales
Optimización en Facebook de contenidos especializados en el sector alimentario. Creación de la empresa LULAF.SL [25]	Lacadena Pascual, Luis	2016	Mencionan la ley, pero no indican el cómo la aplican
Herramienta de análisis de legibilidad de contenidos educativos [26]	Mata San Juan, Henar	2016	No hay datos que proteger
Reducción de dimensionalidad en problemas de regresión [27]	Olivera Fernández-Cortés, Covadonga	2016	No hay datos que proteger
Técnicas de gestión de vuelo autónomo sobre cuadricóptero [28]	Iriz Ricote, Sergio	2016	No hay datos que proteger
Desarrollo de un asistente multimodal educativo para dispositivos móviles Android [29]	Muñoz Moreno, Adrián	2016	No protegen las credenciales
Modificación de estados del teléfono usando la tecnología NFC [30]	Azzahraoui Daoudi, Omar	2016	Mencionan la ley, pero no indican el cómo la aplican

Stifman, un agente de concientización en Second Life [31]	Martín García, Miguel Ángel	2016	No hay datos que proteger
Estudio de solución basada en sistema NoSQL como sustitución del sistema de directorio usado en el Departamento de Informática [32]	Aragonés Tercero, Sergio	2017	Protegen credenciales de usuario
Diseño e implementación de un sistema domótico basado en Raspberry Pi [33]	Santos Senra, Héctor	2017	No hay datos que proteger
Clasificación de patrones de siniestros aplicando técnicas de agrupación y segmentación [34]	Anca Corral, Gabriel	2015	Mencionan la ley, pero no indican el cómo la aplican
Estudio de técnicas de reducción de dimensionalidad para problemas supervisados [35]	Segovia Lasanta, Ignacio	2015	No hay datos que proteger
Análisis de datos aplicado a siniestros de automóviles [36]	González González, Eduardo	2015	No protegen las credenciales
Diseño e implementación de una aplicación para la gestión del cortafuegos de Android [37]	Requena López, Antonio	2016	No hay datos que proteger
C-mulator. Design and development of an educational web application for teaching C language [38]	Uguina Gadella, Lucía	2016	Mencionan la ley, pero no indican el cómo la aplican
Desarrollo de una aplicación de turismo gastronómico para dispositivos iOS y análisis estadístico [39]	San José de Vicente, Martín	2016	Mencionan la ley, pero no indican el cómo la aplican
Localízame for Android: sistema de localización de dispositivos móviles basado en Android [40]	Mertanen Cuní, Nicolás	2013	Protegen credenciales de usuario
Desarrollo de terapias de rehabilitación motora teleoperadas con el robot NAO [41]	Rossignoli Martínez-Vara del Rey, Adrián	2015	No hay datos que proteger

Oracle Business Intelligence for the entreprise [42]	Lee Marco,Hector	2014	No hay datos que proteger
Diseño y desarrollo de una aplicación en Android para la evaluación del rendimiento físico [43]	Rodríguez Jardón, Alba	2015	Mencionan la ley, pero no indican el cómo la aplican
Reconocimiento de tipos de hojas, una aplicación de visión artificial de Android [44]	Sánchez Valhondo, Miguel Ángel	2015	No hay datos que proteger
Gestión de servicios IT mediante códigos QR [45]	Hernández Cassel, Daniel	2014	Mencionan la ley, pero no indican el cómo la aplican
Aplicación de HMMs para clasificar series temporales [46]	Tello Caballo, Faustino	2014	No hay datos que proteger
Editor de vídeo de múltiples fuentes para eventos sociales [47]	Espinosa Montero, José Maria	2014	Mencionan la ley, pero no indican el cómo la aplican
Razonamiento heurístico para fusión robusta de datos en contexto marítimo [48]	Sanchez Faure, Pedro Luis	2014	No hay datos que proteger
Análisis de información proveniente de redes sociales como Twitter [49]	Martín Morales, Soledad	2014	Protegen credenciales de usuario
Un agente en juego Diplomacy [50]	Núñez Pulgar, Santiago	2014	No hay datos que proteger
Simulación de trayectorias de barcos y aplicación al control marítimo [51]	Vázquez Coll, Jaime	2016	No hay datos que proteger
WeSweat, Geolocalización social de deportistas [52]	Muñoz Villar, Carlos	2015	Mencionan la ley, pero no indican el cómo la aplican
Desarrollo de una aplicación multimodal para la consulta de loterías en dispositivos móviles Android [53]	Gómez Sanz, Aitor	2014	No hay datos que proteger
Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana [54]	Arnaiz García, Aitor	2014	Mencionan la ley, pero no indican el cómo la aplican

Jugando al 2048 con Inteligencia Artificial [55]	Sáez Lahidalga, Ignacio	2016	No hay datos que proteger
Categorización de textos científicos mediante aprendizaje automático [56]	Fidalgo Manchón, Luna	2016	No hay datos que proteger
Desarrollo de aplicación de seguridad vial en Android [57]	Urdiales de la Parra, Jesús	2016	No hay datos que proteger
Técnicas de predicción para energía renovable [58]	Mateos Vázquez, Diego	2015	No hay datos que proteger
Monitorización del aprendizaje en redes de neuronas [59]	Pintos López, Roberto	2015	Mencionan la ley, pero no indican el cómo la aplican
Modelado y simulación de trayectorias navales y su representación en Unity [60]	García-Capelo Blanco, Ángel	2015	No hay datos que proteger
Plataforma web basada en la influencia de la climatología sobre el IBEX 35 [61]	Corominas Balseyro, Javier	2015	Mencionan la ley, pero no indican el cómo la aplican
Sistema multiagente para la evacuación de edificios [62]	Pantoja Iniesta, Cristina	2015	No hay datos que proteger
Enfoque de proyecto de implantación de una solución IT para la Gestión del Mantenimiento de Flota en una Empresa Industrial [63]	Mani Ruiz, Rocío	2016	No hay datos que proteger
Valoración de startups con Aprendizaje Automático [64]	García Cazorla, Víctor	2016	Mencionan la ley, pero no indican el cómo la aplican
Sistema móvil de información y guiado [65]	García Herías, Adrián	2013	No protegen las credenciales
Diseño y desarrollo de un cliente y un servidor en JavaScript para gestionar batallas y campeonatos entre agentes inteligentes (JSWARS) [66]	Pérez Ferro, Marcos	2016	No protegen las credenciales

Desarrollo de una aplicación software para laboratorios remotos : control remoto de prácticas (RLF web) [67]	Pacheco Mayayo, Agustín	2014	No hay datos que proteger
Plan de empresa de una compañía tecnológica y desarrollo de su primer producto [68]	Castelo Sagnotti, Fabio Alejandro	2015	No hay datos que proteger
Visión artificial integrada con dispositivos de realidad virtual inmersiva aplicada a videojuegos [69]	Sánchez-Herrero Gómez, Pablo	2013	No hay datos que proteger
Minería de procesos : en ambientes sensorizados [70]	González Escobosa, Carlos Miguel	2016	No hay datos que proteger
Predicción de quiebras empresariales mediante inteligencia artificial [71]	Montero Casarejos, Álvaro	2016	No hay datos que proteger
Cliente Twitter con compresión de datos [72]	Fernández Declara, Plácido	2013	Mencionan la ley, pero no indican el cómo la aplican
Métodos de estimación y análisis de la curva Cupón Cero para el Euro [73]	Escobedo de Pelsmaeker, Alexandra	2015	No hay datos que proteger
Aplicación móvil de geolocalización de mercancía bajo los estándares de comercio electrónico militares Foreign Military Sales (FMS) y STANAG 4329 [74]	Crespo Toubes, Sergio	2014	Protegen credenciales de usuario
Optimización de carteras de inversión mediante técnicas evolutivas y diferentes medidas de riesgo [75]	Antón Aguilar, Alejandro	2015	No hay datos que proteger
Herramienta de gestión para elaboración de cuadros de mando [76]	Fernández González, Patricia	2015	Mencionan la ley, pero no indican el cómo la aplican
Aplicación móvil para la comunicación interna de una empresa [77]	Muñoz Herrero, Diego	2014	No protegen las credenciales

Aplicación móvil para la captura desatendida de datos de sensores en teléfonos inteligentes [78]	Jiménez Fontenla, José Luis	2014	Protegen credenciales de usuario
Predictor en tiempo real de patrones armónicos [79]	Carra García, Pablo	2015	Mencionan la ley, pero no indican el cómo la aplican
Generación Automática de Editores y Repositorios de Evidencias a partir de Modelos de Estándares de Seguridad [80]	Correas Montiel, Elena	2016	Mencionan la ley, pero no indican el cómo la aplican
Fraud prevention through segregation of duties: authorization model in SAP GRC Access Control [81]	Morillejo González, Sandra	2016	Protegen credenciales de usuario
Desarrollo de una aplicación esteganográfica para Android [82]	Pérez Olivares, Sergio	2013	Protegen credenciales de usuario
Análisis de situación y propuestas de mejora para el departamento de Service Delivery [83]	Peral Rodrigo, Alba	2014	No protegen las credenciales
Introducción a la plataforma Arduino y al Sensor ultrasónico HC-SR04 [84]	Martínez Fuertes, Virginia	2014	No hay datos que proteger
Desarrollo de una aplicación de cifrado de imágenes en el sistema Android [85]	Rodríguez López, Iñaki	2014	Protegen credenciales de usuario
Desarrollo de una plataforma social para el suministro colaborativo de piezas de repuesto [86]	Navas Torres, Borja	2015	Protegen credenciales de usuario
Parkineo, aplicación Android para la búsqueda de parking [87]	Romera Alcalá, Iván	2014	Protegen credenciales de usuario
Diseño, desarrollo e implantación de una plataforma empotrada para el control de sistemas robóticos [88]	El Maataoui, Kamal	2014	No hay datos que proteger

Sistemas de construcción de mapas en PDDL para la planificación automática [89]	Caro Herranz, Félix	2013	No hay datos que proteger
Estrategias de diversificación eficiente de carteras e implementación de una plataforma digital de inversión [90]	Pérez Moscoso, Carlos Eduardo	2015	Protegen credenciales de usuario
Interacción humano-robot con el robot REEM sobre el framework RoboComp [91]	Manzano Carrasco, Carlos	2016	No hay datos que proteger
Desarrollo de una aplicación para la gestión de proyectos no gubernamentales [92]	Torres Mendiola, Sofía	2016	No hay datos que proteger
Plan de negocio de empresa basada en Internet de las Cosas y el lanzamiento de un producto: regulador de puerta de garaje por reconocimiento de matrículas de coche mediante Raspberry Pi [93]	Sánchez Checa, Eladio	2014	Protegen credenciales de usuario
SiGUP, sistema de gestión de usuarios para una plataforma distribuída de control de proyectos software [94]	Cabezas Velasco, Oscar	2013	Protegen credenciales de usuario
Técnicas de computación evolutiva aplicadas a la clasificación a partir de monitores de actividad física [95]	Barrio Cerro, María del Carmen del	2016	No hay datos que proteger
Diseño y desarrollo de una herramienta software para la creación de contenidos de realidad aumentada orientada a usuarios finales (end users) [96]	Suárez Esteban, David	2016	No hay datos que proteger

Patrones de paralelismo: una aproximación basada en bibiliotecas génericas [97]	González Sánchez, Víctor	2016	No hay datos que proteger
Gestión de rutas y toma de decisiones en el entorno de simulación STI-SIM [98]	Zamora España, Víctor Manuel	2016	No hay datos que proteger

Tabla 75: Colección de TFG analizados

ANEXO III: Manual de usuario

El módulo propuesto permitirá fomentar el conocimiento de los desarrolladores software respecto al uso de buenas prácticas para la autenticación de usuarios. Este módulo proporcionará una solución básica que permitirá que los desarrolladores tengan una herramienta para autenticar y se puedan "olvidar" de la protección de credenciales.

La autenticación de los usuarios es en algunos de los sistemas, un agujero de seguridad que puede ser aprovechado por atacantes para extraer los datos personales. Por lo tanto, es necesario que los desarrolladores se mentalicen de seguir una serie de buenas prácticas.

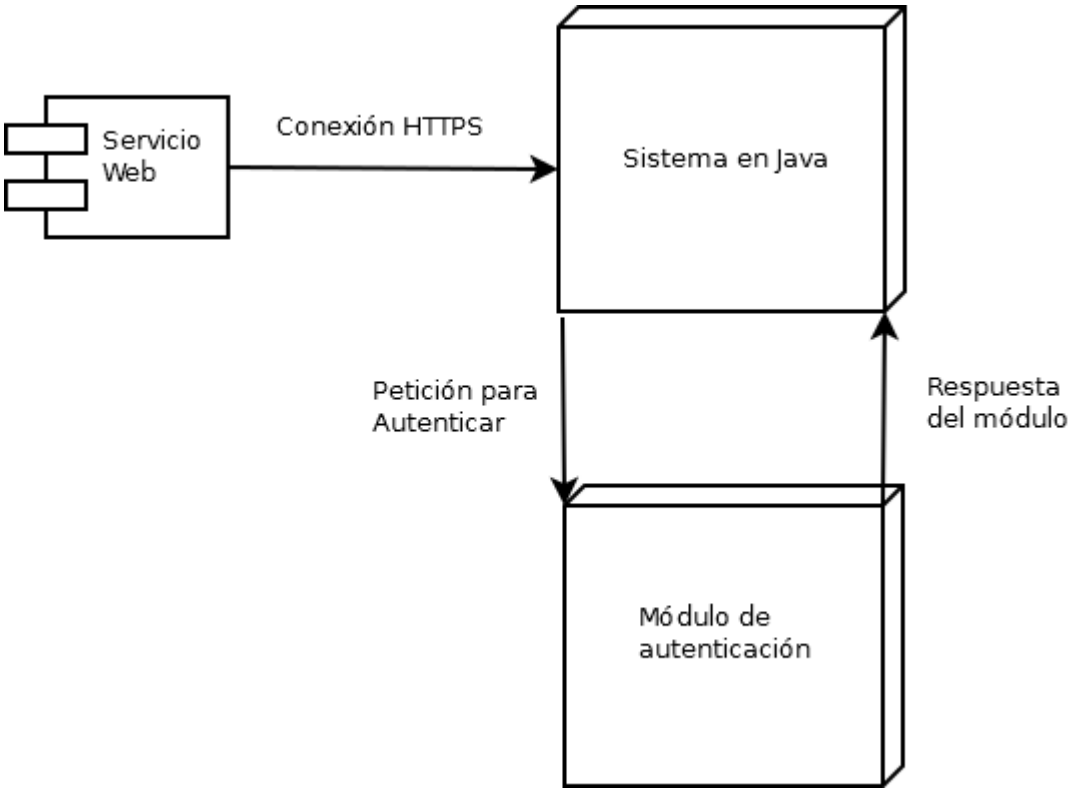

Ilustración 43: ANEXO III: Diagrama del sistema con módulo de seguridad

El módulo puede ser utilizado en cualquier servicio Web implementado en Java. Cómo podemos ver en la ilustración anterior, el servidor se comunicaría a la hora de la autenticación o cualquier funcionalidad que tenga influencia en el manejo de los usuarios. Por lo tanto, en caso de tener un servicio Web que no tiene autenticación, seria sencillo implementar este módulo basado en ESAPI de OWASP.

La librería contiene una serie de buenas prácticas que permiten securizar un sistema de manera sencilla. Entre estas buenas prácticas se encuentra el almacenamiento de contraseñas mediante función resumen con un determinado protocolo, que la librería proporciona facilidad para cambiar (agilidad criptográfica), y un determinado SALT que también se puede modificar.

Además es totalmente configurable para el número de intentos de contraseña antes de bloquear el usuario, permite seleccionar diferentes algoritmos de cifrado (AES o DES por ejemplo) para el intercambio de datos si fuera necesario. Los datos de entrada son

validados también para que cumplan requisitos mínimos de seguridad y generación de Tokens y manejadores de sesión para la correcta identificación de los usuarios.

En sistemas más complejos te ayuda a prevenir los ataques Cross Site Scripting y la denegación de servicio. También, permite un manejo de roles, por lo que como comentamos en otros puntos del TFG se podría crear un sistema más complejo en el que existan varios roles a parte del administrador ya creado.

A continuación explicamos la implantación del sistema y como utilizar la aplicación. La implantación del sistema se encuentra presente dentro del TFG en el apartado 6.2 IMPLANTACIÓN DEL SISTEMA, pero es copiada de nuevo por la posible distribución del anexo de manera separada.

Implantación del sistema

Para la implantación del sistema se realizara un fichero ZIP con todo lo necesario para que el sistema funcione correctamente. Este ZIP estará formado por el proyecto que habrá que importar en eclipse, un almacén de claves, una carpeta con la configuración de ESAPI y por último, una carpeta con el servidor de Tomcat, ya que tiene ciertas modificaciones.

El primer paso para la implantación del sistema es copiar el almacén de claves (.keystore) y la carpeta de configuración de ESAPI (.esapi) en la ruta C:\Usuarios\[Nombre_Usuario]. A continuación, se muestra unas rutas de ejemplo.

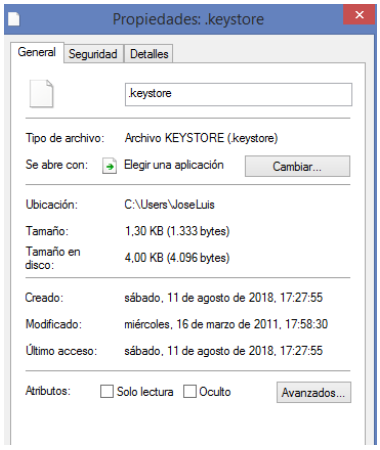

Ilustración 44: ANEXO III: Propiedades .esapi

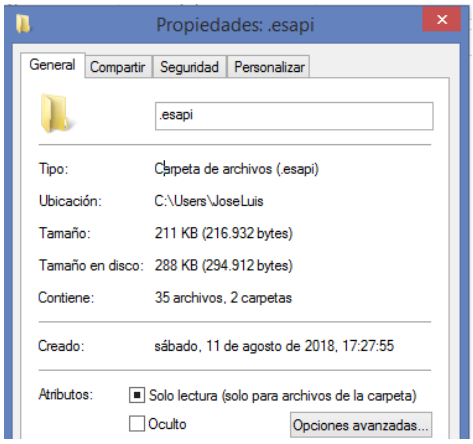

Ilustración 45: ANEXO III: Propiedades .keystore

Dentro de la carpeta de configuración de ESAPI se encuentran varios ficheros y carpetas. A continuación se mencionan los más importantes y los que pueden necesitar modificarse dependiendo del sistema en el que se incluya.

- ESAPI.properties: Contiene las propiedades de la librería. Por ejemplo el método de cifrado, el método de HASH, numero de fallos al iniciar sesión… En caso de que un sistema concreto necesite un tipo de cifrado distinto, se podría modificar por ejemplo.
- validation.properties: Contiene las validaciones del sistema. Es decir, expresiones regulares que determinen el formato de ciertos datos de entrada.
- users.txt: Es el fichero en el que se almacenan los usuarios y su información según se van registrando en el sistema.
- Carpeta fbac-polices: Contiene distintos ficheros que permiten el bloqueo a información a través de roles. Por ejemplo, el fichero URLAccessRules permite o bloquea el acceso a ciertas URL dependiendo del usuario. Por ejemplo, nuestro sistema la utiliza para que solo el administrador pueda acceder a la pantalla de administrador.

Cómo hemos mencionado anteriormente, en la carpeta de configuración de ESAPI hay más carpetas y ficheros, pero los mencionados son los que serán útiles en caso de modificarse en nuestro sistema.

Tras esto, se puede importar el proyecto en eclipse. Para ello se importará la carpeta TFG en eclipse como proyecto existente. A continuación se muestran capturas de cómo se realizaría.

Pulsamos en importar y seleccionamos Existing Projects into Workspace.

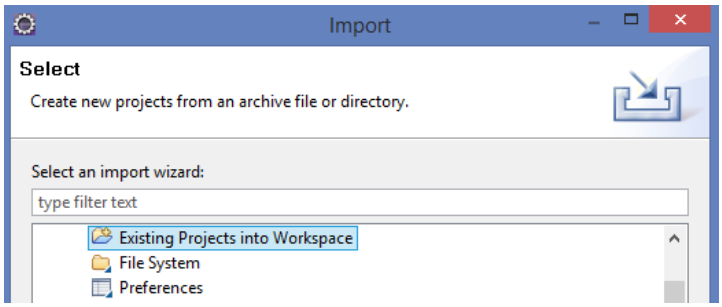

Ilustración 46: ANEXO III: Importar proyecto

Tras esto, buscamos el directorio raíz, que estará dentro de la carpeta tfg y tendrá el mismo nombre, TFG, cómo podemos ver en la captura. Una vez seleccionemos la carpeta y pulsemos finish, el proyecto se incluirá en eclipse.

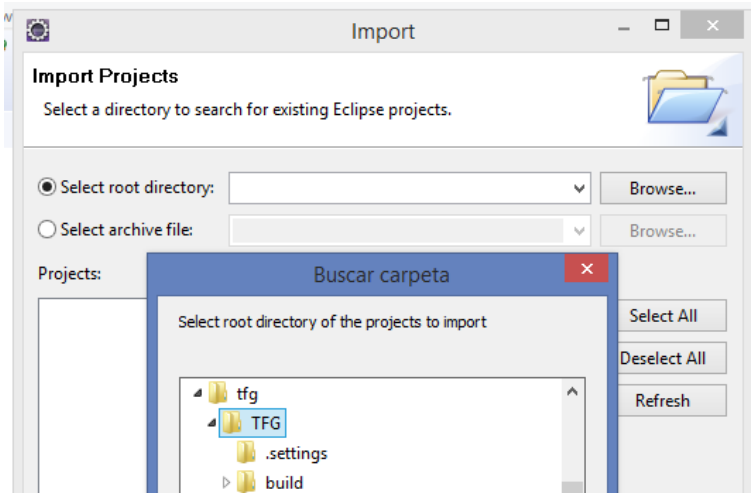

Ilustración 47: ANEXO III: Seleccionar raíz de proyecto

Una vez se ha importado correctamente el proyecto en eclipse, se podrá modificar el código en el caso que el sistema a implementar lo requiera.

Por último, faltaría añadir el servidor. En nuestro caso, proporcionamos apache-Tomcat 9.0.10, ya que el proyecto tiene sus propias configuraciones del servidor. En caso de querer utilizar otra versión de Tomcat no habría problema siempre y cuando se modifiquen los ficheros de configuración de manera que queden similares a los proporcionados. En caso de utilizar otros servidores, por ejemplo Glassfish, habría que adaptar al igual que en Tomcat las configuraciones del servidor, para que permita su correcto funcionamiento

Para añadir el servidor en eclipse, habrá que moverse en la parte inferior a la pantalla de servers y pulsar con el botón derecho para crear uno nuevo.

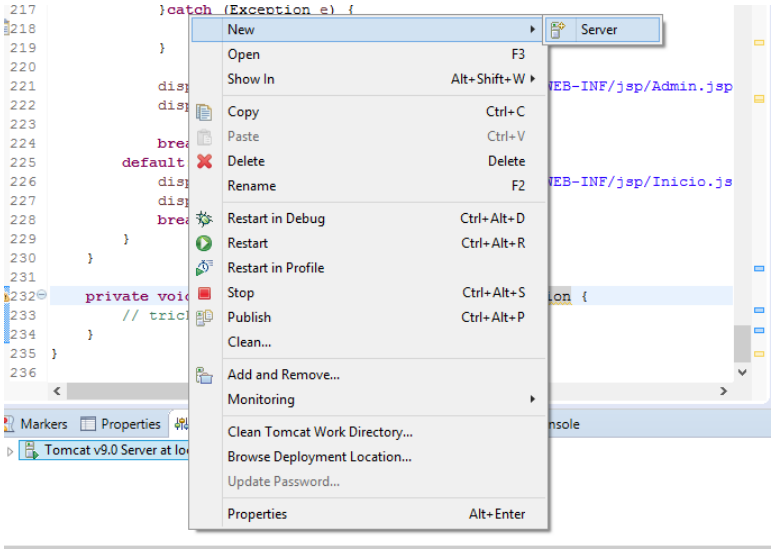

Ilustración 48: ANEXO III: Añadir servidor

Una vez pulsado el nuevo servidor, elegimos el tipo, en este caso Tomcat v9.0.

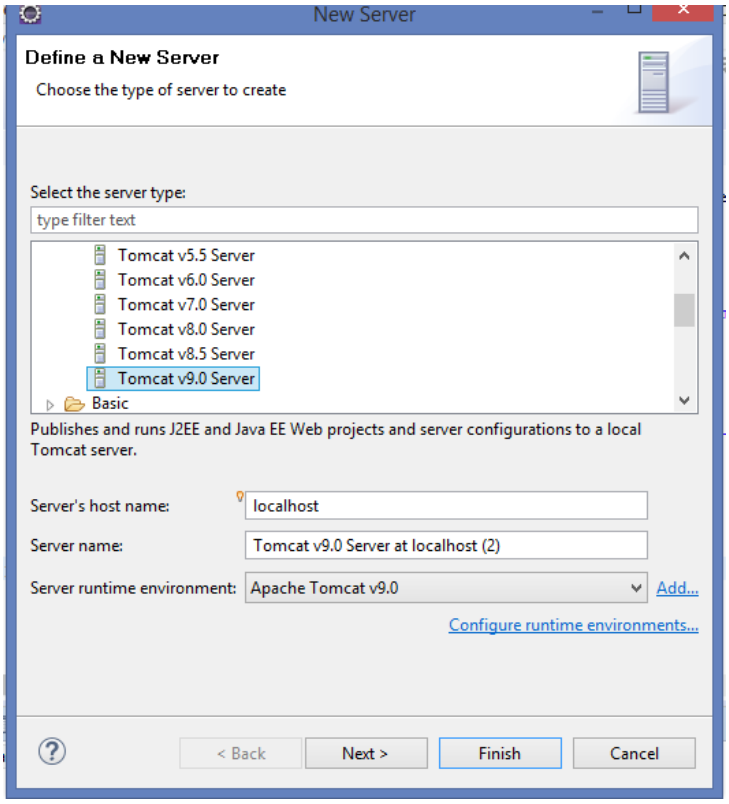

Ilustración 49: ANEXO III: Seleccionar servidor

En caso de no venir por defecto la ruta en la que tenemos Tomcat, habrá que pulsar Add e introducir la ruta en la que lo tenemos cómo vemos en la imagen inferior. Tras esto, pulsamos en finalizar.

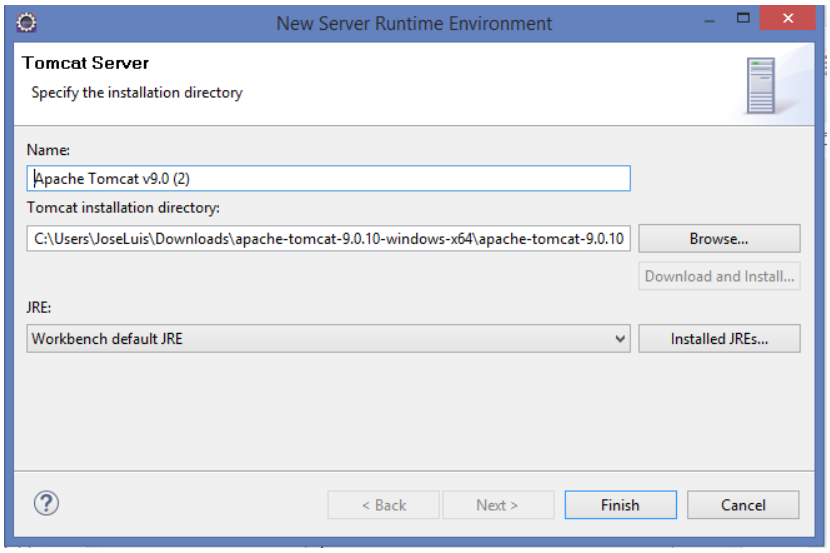

Ilustración 50: ANEXO III: Ruta del servidor

Por ultimo añadimos el proyecto a la parte de Configured, y ya estará disponible para lanzarse a través del servidor. Por último damos a finalizar.

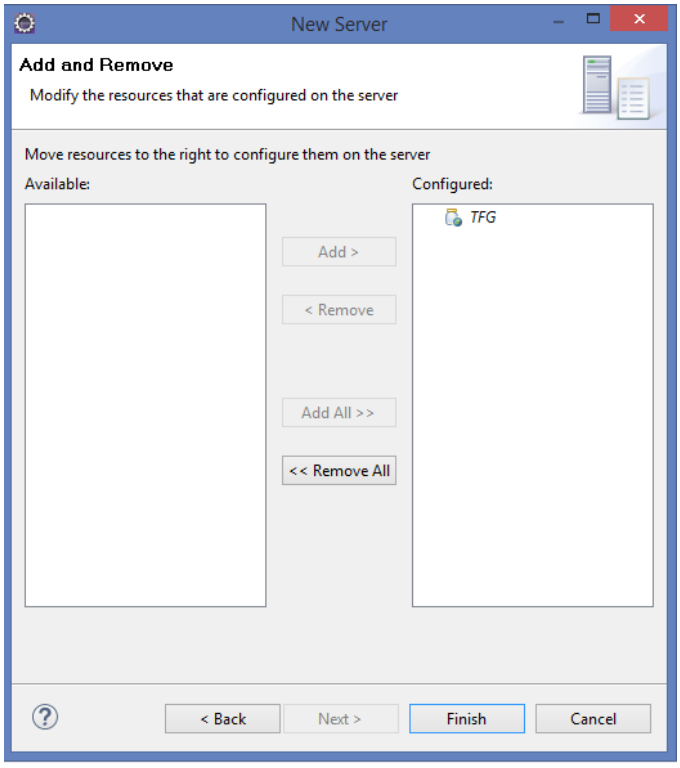

Ilustración 51: ANEXO III: Añadir proyecto al servidor

Con esto el sistema estaría completamente operativo en cualquier entorno y se podría introducir en cualquier sistema. En el caso de necesitar modificaciones en el sistema, se podrían realizar desde eclipse o en cualquiera de los ficheros mencionados anteriormente.

Cómo utilizar la aplicación

A continuación explicaremos de manera detallada por pantalla, que puede realizar el usuario de la aplicación, para que pueda utilizar de manera sencilla la aplicación.

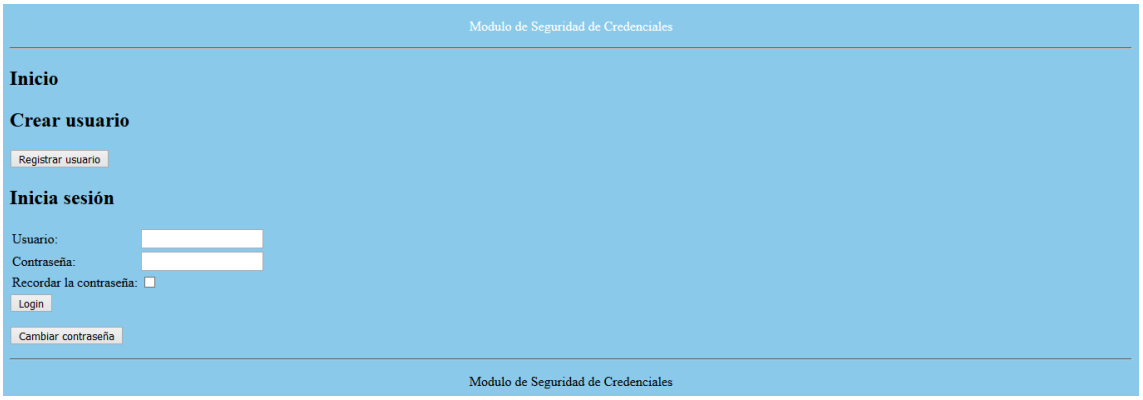

Ilustración 52: Manual de usuario: Pantalla de inicio

En la pantalla de inicio, como podemos observar, se encuentran las funcionalidades de crear usuario, iniciar sesión y cambiar la contraseña. Para diferenciar entre roles, vamos a crear un usuario sin rol. Por defecto viene creado un usuario administrador, cuyo usuario es admin y su contraseña Admin001.

Usuario no administrador

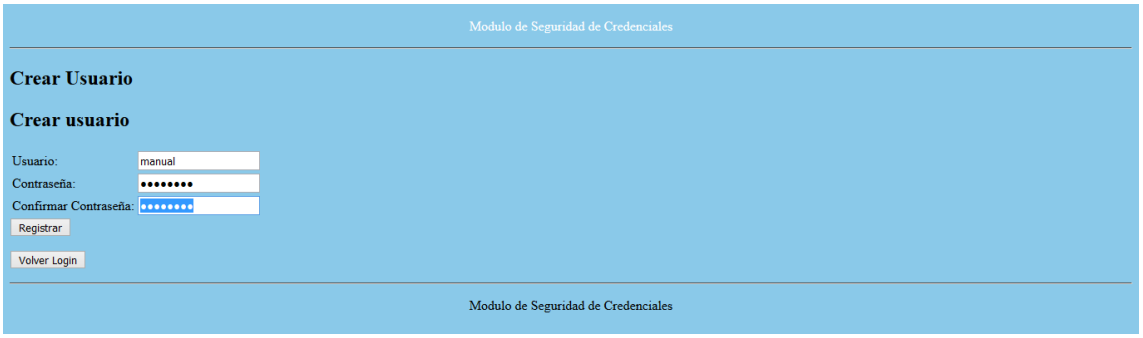

Ilustración 53: Manual de usuario: Registrar usuario

Una vez en la pantalla de inicio, creamos un usuario, para ello en la pantalla de inicio mostrada anteriormente pulsamos en crear empleado. En este caso el usuario será manual y la contraseña Manual01. Una vez pulsamos en registrar, aparecerá un mensaje de que el usuario se ha registrado y deberá pulsar en volver al login.

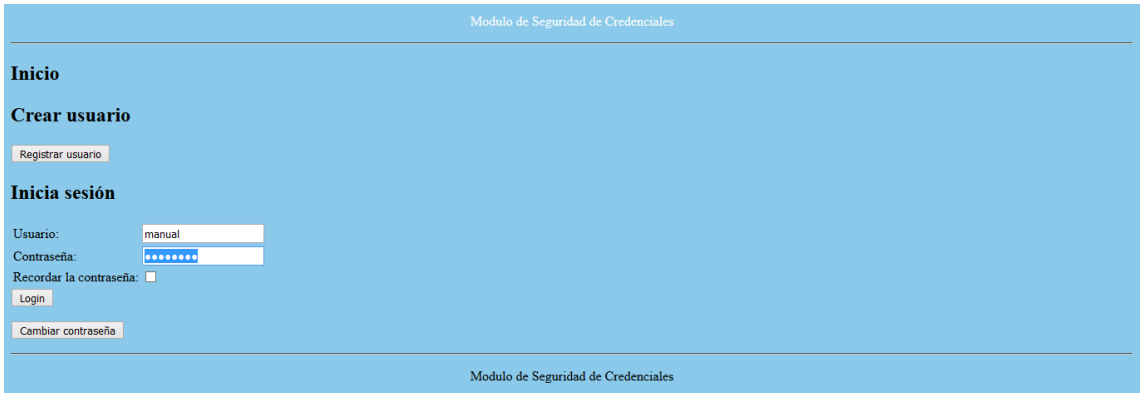

Ilustración 54: Manual de usuario: Introducir usuario y contraseña

Una vez tenemos el usuario registrado, introducimos usuario y contraseña y pulsamos en Login.

Ilustración 55: Manual de usuario: Pantalla de conectado

Una vez se inicia sesión, la pantalla que verá el usuario tras el inicio de sesión será esta. En caso de actualizar la pantalla, los datos de usuario desaparecen, ya que solo es una información tras el inicio de sesión. En esta pantalla, se permite cerrar sesión o borrar el usuario permanentemente del sistema. También se permite el acceso a la pantalla de administrador, pero cómo en este caso no tiene permisos suficientes, el botón no hará nada, únicamente actualizará la pantalla.

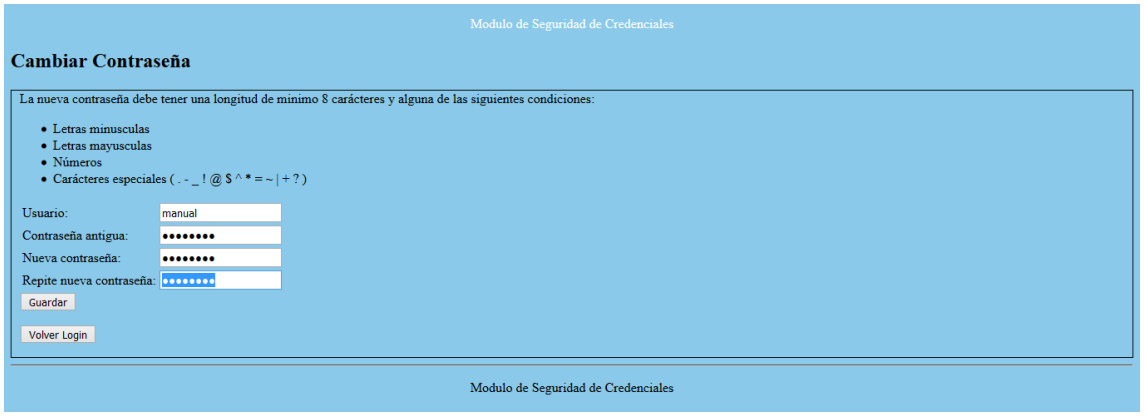

Ilustración 56: Manual de usuario: Cambiar contraseña

Desde la pantalla de inicio, se puede cambiar la contraseña. Para ello se debe rellenar el formulario de la imagen anterior y pulsar en guardar. Una vez se realiza el cambio correctamente, se deberá volver al login con el botón Volver Login.

Usuario administrador

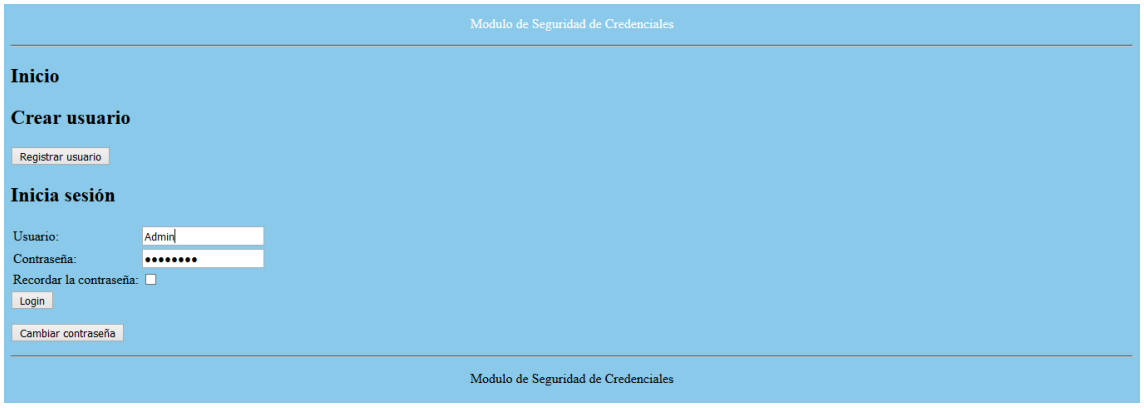

Ilustración 57: Manual de usuario: Iniciar sesión con administrador

Accedemos al sistema con la cuenta de administrador mencionada anteriormente.

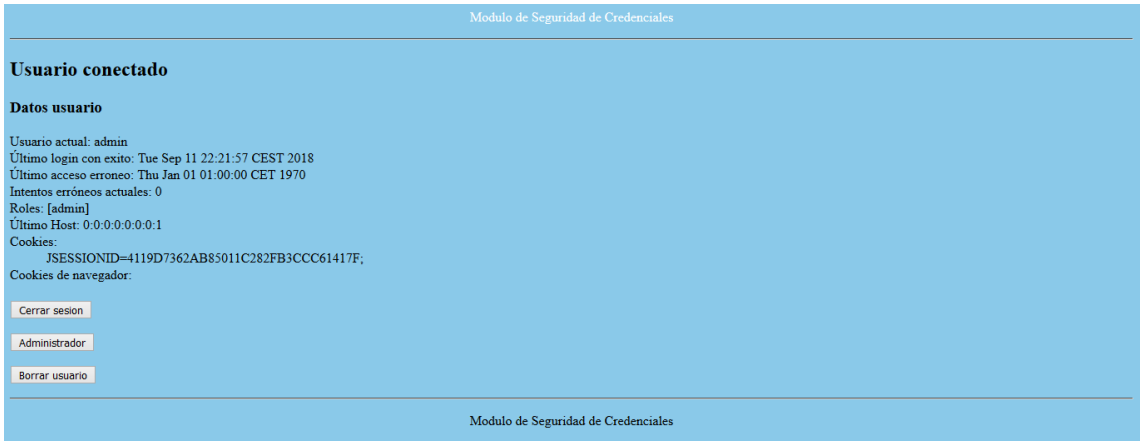

Ilustración 58: Manual de usuario: Usuario administrador conectado

En este caso, el usuario administrador puede entrar a la pantalla de administración, a diferencia del usuario mencionado anteriormente. A continuación vemos las funcionalidades que puede realizar el usuario administrador.

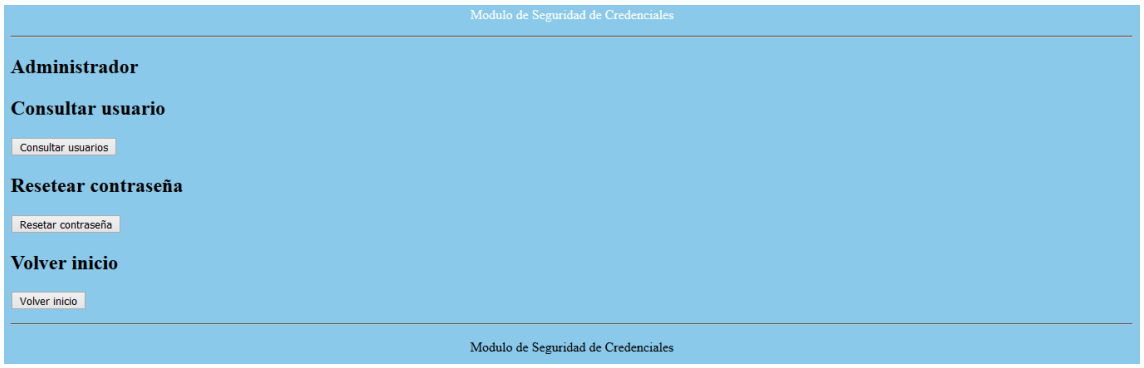

Ilustración 59: Manual de usuario: Pantalla de administrador

El administrador, podrá consultar usuarios o resetear la contraseña. Si pulsamos en consultar usuario, pasamos a la siguiente pantalla.

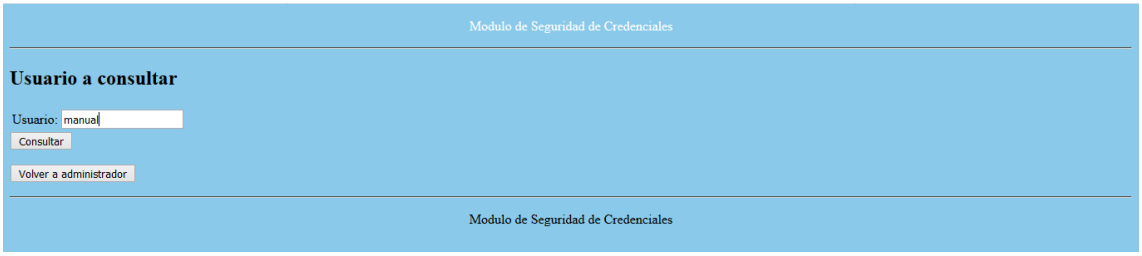

Ilustración 60: Manual de usuario: Consultar usuario

Una vez pulsamos en consultar usuario, nos lleva a esta pantalla, en la que indicamos el usuario que queremos consultar. Una vez introducimos el nombre, pulsamos en consultar y consultamos el usuario.

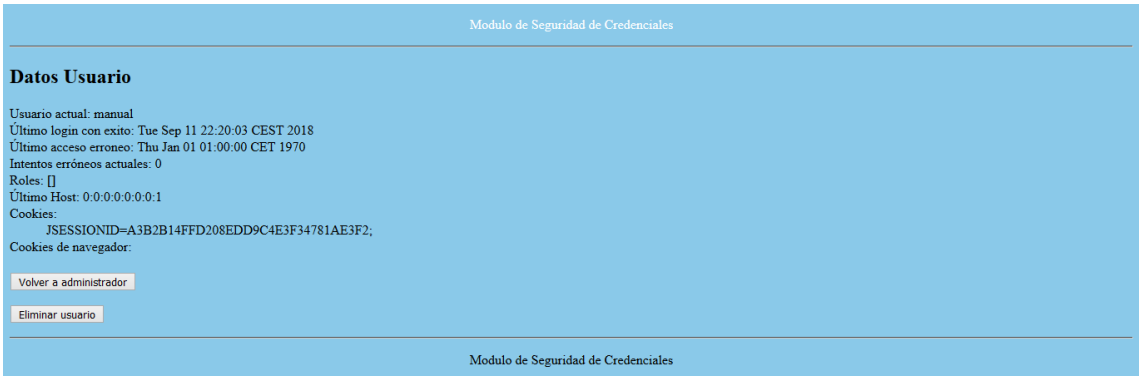

Ilustración 61: Manual de usuario: Usuario consultado

Una vez pulsamos en consultar, se extraen los datos del usuario y se muestran por pantalla. Además, se puede eliminar el usuario del sistema pulsando en el botón de Eliminar usuario. En caso de querer volver a la administración, habrá que pulsar en volver a administrador.

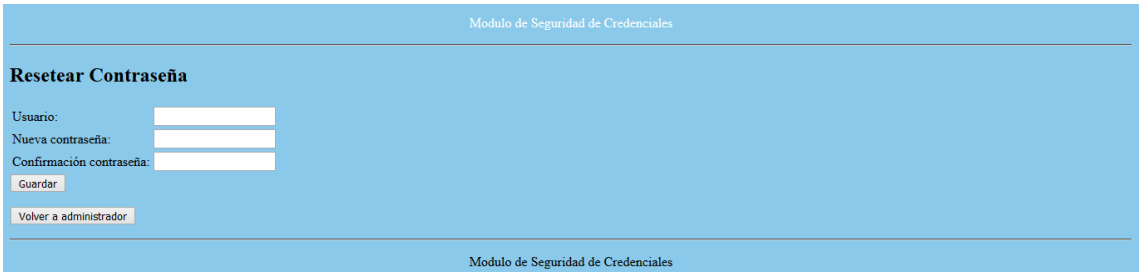

Ilustración 62: Manual de usuario: Resetear contraseña

Por último, desde la pantalla de administrador, se puede acceder a la pantalla de resetear contraseña. En ella, se deberá rellenar el formulario con el usuario a la que se quiere resetear la contraseña y pulsar en guardar. Cabe destacar, que la contraseña tiene que cumplir los requisitos de seguridad.

ANEXO IV: Trabajo en inglés

Abstract

In this report we explain the development of a system with the main objective of protecting the credentials of the users. It has been designed to be highly integrated in the different systems, acting more as a module than a standalone application. In order to achieve this goal, we have analysed several protocols and types of authentication that have helped us to acquire the knowledge to appropriately develop the system.

In addition, an investigation of the previously published Final Year Projects (FYP) is carried out thanks to the records of the University library online repository. This study made us to realize how important is the protection of the user credentials, and its presence in other systems.

In this point, we had an idea of the importance of the protection and some of the approaches of other developers, so we started the design and analysis of the system. This design consisted in a security module implemented in JAVA using the ESAPI library, which allow us to perform a secure maintenance of the users. Once implemented, a set of test was defined to prove how it meets the requirements.

Finally, the report includes the time and budget planning followed to achieve the realization of this project, along with a simple user manual that explains how to actually use the module.

Introduction

Nowadays, we all have some kind of personal information stored somewhere we do not own, such as in the case of social networks, the cloud and different other applications which make use of our personal data and store it within external computers. These days, people tend to allow more and more data to be distributed all around the internet, so in terms of security, this is becoming a problem. It is not only that users are insecure when they perform a simple connection to the internet, but the smartphones we carry with us daily keep lots of personal data that is subject to vulnerabilities, making our data accessible to unauthorised parties.

Thus, smartphones constitute yet another way for malign hackers to dig some interesting data about us, which eventually they can sell or use to perform extortion over sensitive information yielded from security issues with applications.

In order to prevent these vulnerabilities from happening, it is mandatory that our systems have an appropriate protocol for treating personal information of users. The idea is to apply mechanisms and procedures that guarantee the security of the data handled by our system.

However, there are still too many systems that do not perform this data treatment at all or do it incorrectly, so there are plenty of programmatic tools that ease the implementation of authentication in these systems, to avoid the aforementioned problem of unauthorised access.

Credentials are the kind of personal data that is often target of security attacks, these correspond to information that enables the identification of users towards some

system or application. The simplest and most widely used approach to implement authentication is the combination of username and password we are all used to dealing with. Consequently, if we keep the information regarding username and password safe, then, we would be protecting the confidentiality of the personal data held within the system these credentials are used.

Motivation

According to my previous experience in the Computer Science Degree, I know that there have been published many Bachelor projects which show applications that implement authentication, nevertheless, these applications do not usually treat personal information as carefully as they should. The main problem I have spotted with these applications is that generally, personal data is stored in a database in clear text or in some other cases, the author of the project does not even provide a solution for data storage or management.

I believe security must be an issue concerning all kind of applications, for sure including final degree dissertations. It is crucial to design and implement any system, even the simplest one with security in mind, as an inherent requirement of final application. We may be handling personal information in a program written for our dissertation, but we never think that our code may be used as base for similar enhanced projects and future users or supporters of the system might suffer from our carelessness.

This stress in security would be necessary for the students to realise of the criticality of security in barely any computer system, and would serve as a great knowledge base to later face their professional career outside university. Of course, information security is an obscure topic that not everyone is familiar with, so it would not make any sense to tell all undergraduates to implement their own security mechanisms in applications which are not targeted to this field on study.

This is why I decided to develop myself a small security module that implements simple authentication, which is intended to be reused in future projects in which time constraints negatively the security of the final system delivered.

Objectives

In this paper I will be covering the state of the art regarding approaches to protect credentials, trying to provide a relatively high variety of solutions to secure authentication.

Next, a set of bachelor dissertations from previous years will be presented and used as a landmark to gather information on how well data is protected, spotting problems that the present dissertation will focus when it comes to the development of the security module proposed here. In addition to the module, this document is aimed to provide a set of learning materials to ease the reuse of the module in later work.

The main purpose of this work is to provide a solution for credential protection, to be used in Bachelor dissertation projects and implemented in the Java programming language.

In order to provide an accurate solution that is tailored for this purpose but flexible to adapt to any final dissertation app, we need to follow a procedure that enables the

clarification of the final requirements the system must comply with, so that it fits any TFG.

The process I will be following along this document is:

1. State of the art analysis regarding credentials protection.
2. Gathering of data and performance of statistic studies from previous year's dissertations.
3. Essay depicting the possible solutions based upon the former context studies.
4. Implementation of a standard solution that is flexible enough to adapt to most final degree projects.
5. Creation of documentation for the implemented tool, which will ease the use of the module in subsequent work.
6. Discussion regarding the importance of using tools that protect credentials.

Structure of the paper

The document will begin showing a brief introduction where the reader will find a summary of the work that is to be done and described along this paper. Moreover, such summary will be accompanied by a list of objectives that are aimed to be achieved in this project, along with some comments regarding the motivations the author has to carry it out.

Later, the paper will present a state of the art study, providing an overview of approaches that were taken from the past to the present to protect credentials. This is an important part of the paper, since having a knowledge base about how other people approach an issue would guide any developer to find a better solution to the problem, avoiding paths that others have tried and failed too.

Once we are done with the state of the art study, I will set a statistic study off, extracted from the analysis of previous years dissertations. These numbers will provide us with an idea about the role that our system would have if integrated with those former projects, and will conclude whether this secure module is as promising as it seemed when I came out with the idea.

Next, a list of formal requirements will be introduced, starting with user requirements (extracted from our analysis of previous projects), we will obtain software requirements, which will be the ones our system will have to implement to comply with the user requirements, accordingly. We will be taking into account the environments in which the system will operate.

After requirements are depicted, we will go through the design of the module. In the design section, the architecture of the system will be described, use cases will be pictured and special care will be taken to offer a simple interface that allows an easy use of the system.

In the following section, we will briefly describe the implementation process, in order to clarify the key details and milestones involved in the development and a manual regarding how to integrate the present module and any other app is also included.

After describing the application as a whole, we will evaluate it, in order to check compliance with the initial requirements. In this part, the reader will find the project

management details, which comprise methodologies used, budget and time planning as well as the legal regulatory framework that affects the project.

In the next stage, I will be discussing the results and conclusions obtained from the development of this application and future work will be plotted.

Finally, a user manual will be presented, to ease the reuse of the module. This software product documentation will contain screenshots and simple instructions to make the process of integration as painless as possible.

State of the art

User credentials are sets of information that allow identification of users. These are used in authentication processes in software systems, enabling this way access to the authorised parties or users to information held within the app in a secure manner. Authentication consists in confirming that you are who you claim to be, which is, of course, a crucial requirement in any app data transaction. Due to the potential damage that can be produced to users if malicious hackers impersonate them, it is a must to implement shielded authentication mechanisms.

The protection of credentials has been an issue for programmers a long time now, but at least there has been an improvement with time regarding the security level achieved by newer authentication approaches and mechanisms. Considering the oldest ways of implementing authentication, using user and password, we now have complex credential mechanisms, such as those using biometrics. Below, I show a brief list of approaches that have been used as authentication mechanisms.

- ID and password: this is the simplest method to check whether a user is who is claims to be. This is, for instance, the method that Aula Global uses.
- Key Exchange: it is a more complex method, in which public and private key cryptography is involved to secure the authentication of entities involved in a transaction.
- OTP: "One-time passwords", the password is set for a period of time and only allows a single session.
- Biometrics authentication: makes use of the physical unique characteristics of users to check their identity. Some biometric mechanisms are the following:
    - Fingerprint: authentication is achieved by means of the fingerprint, which has a very low probability of being identical to that of another person.
    - Iris: as it is yet another unique part of our body, its analysis can be used to authenticate.
    - Face recognition: according to a set of parameters extracted from the analysis of the face characteristics, one person is differentiated from another, allowing authentication.
- Active RFID: RFID technology and tags allow authentication based upon some item that only the authorised user has, since tag and server of the RFID architecture talk to each other to grant access to it.

However, for the purpose of this dissertation, we will be focusing exclusively in PAP authentication systems, that is, user and password authentication systems, which are the most widespread respect to previous years' dissertations.

To provide a solution to the vulnerabilities of this authentication scheme, we will need to spot the actual vulnerabilities first:

- Easy-to-guess passwords: no matter how well you protect passwords from being read in cleartext, even someone with no hacking skills can guess a poor password, thus, we must promote the use of strong passwords.

- Default passwords: some systems provide an initial password to their users, warning them to change it as soon as possible, which by the way doesn't happen very often. These passwords often follow a generation pattern and can be compromised easily.

- Shared passwords: in systems that do not support several logins, users are forced to use the same password to access to everything. This causes lots of trouble, since the real user is never authenticated, the device itself is authenticating a fictitious user, which we cannot know whether it corresponds to the real one or not. Moreover, even in the case that the original user does not use the device anymore, the device will probably keep the information of the user still, so unauthorised access to personal data may happen.

- Raise of credential use all over the internet: as the Internet is in its peak, nowadays, more an more people must have credentials to access some application of their phones or PCs. This implies a raise in the amount of confidential information that travels through the network. As many users use the same credentials for several sites or apps so that they can remember them, as soon as one site gets leaked, the disclosed credentials of one users enable access to all of its accounts.

- Privileged accounts: most systems incorporate administrator accounts which allow total access capabilities over the system. These are a great tool to control the correct flow of your application and solve problems the users may come up with, but a bad use of these accounts may lead to the compromising of the whole system [1].

In order to secure our systems and make sure they handle information appropriately, the impact of the aforementioned vulnerabilities must be minimized. In general, the effort put into securing a system is not sufficient, always inferior to the ideal standards, since securing a system involves a high monetary cost. Since many standards have been defined to provide with essential measures to be taken for applications of any kind, systems should be at least implementing enough security so that these directives are fulfilled.

On the other hand, implementing security mechanisms usually constitutes a drawback regarding system management, its development timespan and might yield to more effort required to build and maintain an application.

If we get deeper into the matter, late efforts have been proposing stronger authentication mechanisms, which add extra components to the classic user and password tuple. A popular example is MFA, Multifactor Authentication, which requires several authentication levels to verify the legitimacy of a software system transaction.

The main advantage present in MFA methods is in fact they use credentials which are independent from each other. This is great in terms of security, because even in the case that user and password are compromised, the attacker will still be unable to access the user personal data, since he/she is lacking the second layer credentials. This approach does not actually protect in a different way credentials, but instead, it stacks them in the authentication process so that it is harder for an attacker to penetrate the system.

Even though companies and developers have total knowledge about the fact that their systems need to do something to protect users' sensitive information, they tend to ignore these problems and only face reality when some hacking incident involves their application. Maybe they should change their mind, since lately there is a raise in the number of cybersecurity attacks, which actually make use of the ignored vulnerabilities the applications live with.

In reality, some efforts have been put to solve this data leakage happening right now. IAM frameworks, Identity and Access Management frameworks are becoming popular these days.

IAM frameworks are tools that ease the management of user identities on a system. In other words, they simplify the work required to secure access of the users to the system by providing an independent module that is solely in charge of protecting the credentials and identities of users. These frameworks allow handling the permissions each user has using a role-based assignment of access capabilities, to be granted by the administrators of the system

IAMs include single-sign-on systems, multifactor authentication schemes and access management support. These technologies also offer the possibility to store in a secure manner the identity and profile information of a given user, as well as the functions targeted at guarantee that only the strictly necessary data is shared with the requestors.

On top of that, these systems support capturing and recording the logins performed by users, allowing this way their supervision. IAMs shall also simplify the process of configuration of user profiles, minimizing the possible security vulnerabilities introduced when it comes to perform this configuration manually. It is also important to provide mechanisms that ease the modification of the data. [2]

In the near future, with the growth of the IoT (Internet of Things) paradigm, a greater number of devices and systems will get a connection to the internet. All of these systems will carry with them personal information, which will place a perfect scenario for malicious hackers, which will be provided with yet another data leakage source. Therefore, we will be soon see appearing new security modules applied to devices laying within this paradigm.

Another security problem we will be facing in the future is cloud storage authentication. Nowadays most systems use protocols such as O-Auth 2.0 (open authorization), which allows simple approaches to prevent unauthorised access to web services or applications over the network.

Previous Final Degree Dissertation statistics.

So that we can have a glimpse of the impact our security module may have in computer science dissertations, I have developed a small study of previous years' bachelor papers so that we can numerically estimate how many of these care about security of user credentials.

The following table summarizes some conclusions from data gathered from final degree dissertations obtained from the e-archivo:

	2017	2016	2015	2014	2013	Total
Don't protect credentials		3	1	2	1	7
Mention legal regulations but never tell whether they apply it		9	7	3	1	20
Do protect user credentials	2	1	2	6	3	14
There's no data to protect	1	18	9	8	2	38
Total	3	31	19	19	7	79

Tabla 76: TFG Statistics

As we can see, the amount of papers that actually claim to have their user credentials protected are very few, from a sample of 79 projects, only 14 of them tell how they protect these data.

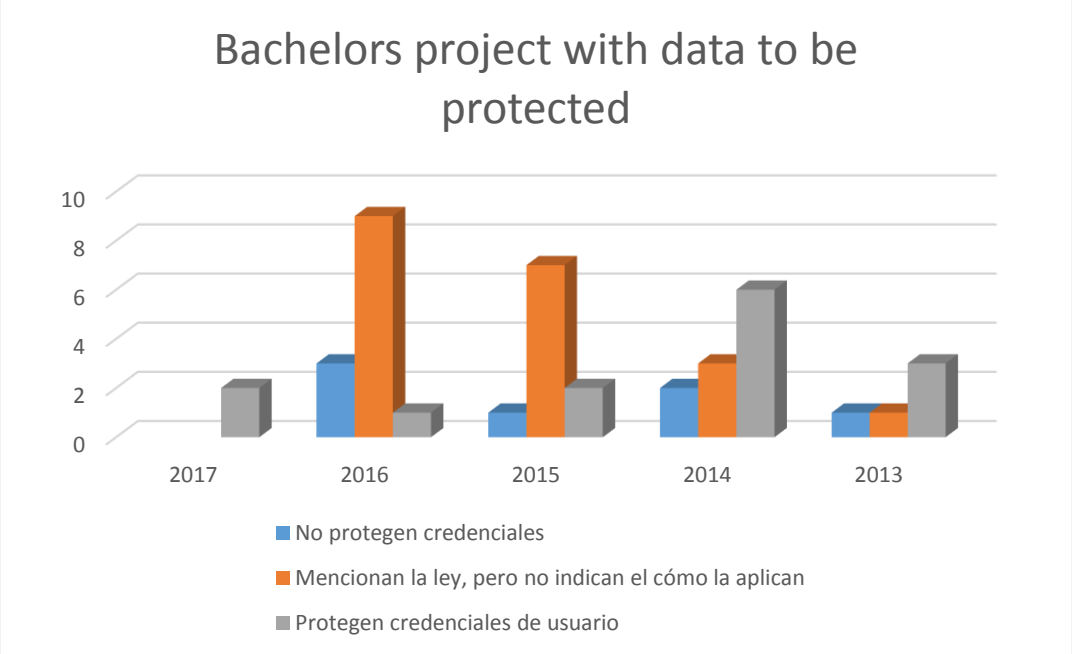

Ilustración 63: Statistics of projects with data to be secured.

If we focus in projects that do have data to be protected, we can observe that every single year, students show they know they must apply the LOPD (Spanish Law of data

protection). However, a very small percentage explains the way they actually apply these regulations to their projects. That is, they do not often tell which kind of encryption mechanism they have used, nor which is the data they did protect according to the different security requirements stated in the regulation, which would suffice for us to realise they comply with the law.

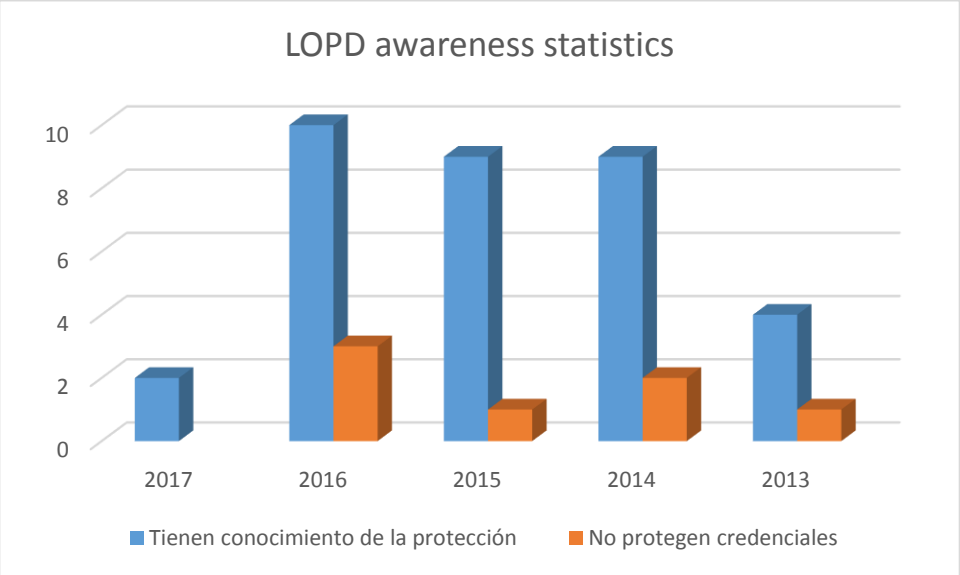

Ilustración 64: LOPD awareness statistics.

As a positive stress, we can outstand the fact that most students are aware of the need to protect user data. We may also consider the possibility that they forgot to write the way they protected data in their dissertation.

Analysis

To develop software components, it is necessary to analyse the problem to be solved carefully, so that we can define the real needs to be satisfied. In order to do so, this section will cover distinct aspects on how the lifecycle of software development can be approached, the proposed solution, user requirements, software requirements…

Software Lifecycle

To develop the module, we have chosen to use a cascade approach. This is the simplest software lifecycle approach but adapts perfectly to the kind of system we aim to build and cover in this paper. The different phases will be overcome in a sequential order. The name of the cascade lifecycle comes from the actual positioning of the different phases in the scheme showing the process.

This lifecycle has been chosen since the software requirements are pretty clear from the very beginning, so they are not expected to change in the future, once the development has started. The solution can be improved later in the maintenance phase of the project.

Features of the proposed solution

With the help of the DIA tool, we show below a small diagram plotting a high abstraction level description of the module we cover in this dissertation.

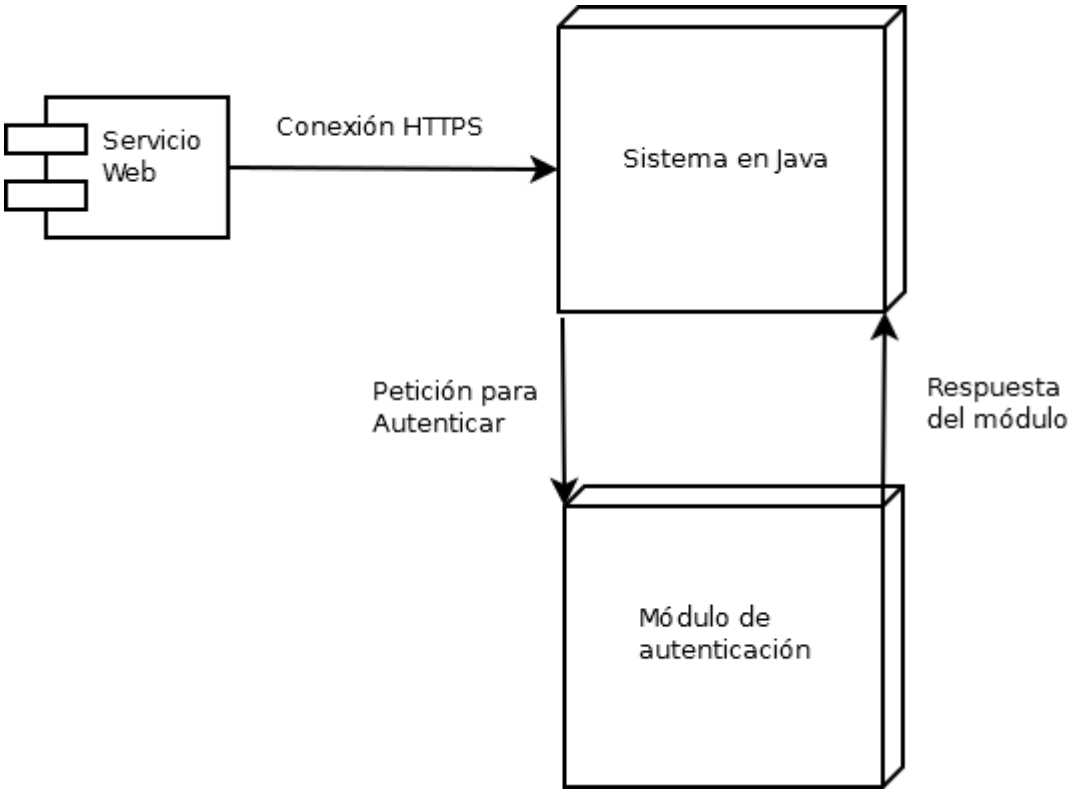

Ilustración 65: Component diagram of the security module

As we can see in the figure, the web service would be accessing our module when it aimed to perform operations related to users, such as the creation of a user, deletion of a user or its authentication.

The authentication module to build must present a set of characteristics compliant with key objectives of the application, mentioned in the "Objectives" section. The features to be supported by the system are:

- The system must allow to record users within the system.
- The system must allow to log in to a user that has been previously recorded in the module.
- The system must allow to modify the user's passwords.
- The system shall allow terminating a session to a user which has an active session in the system.
- The system must allow to delete a user from the system.
- The system must integrate easily with other software systems.
- The system must comply with the software security regulations regarding data protection.
- The system must allow the update of the authentication protocols it uses in case these become, eventually, obsolete.

- The system must be fairly easy to use to users, providing an intuitive user interface.

Chosen solution

In the section "Study of the possible solutions" we dug into the different methods by which we can implement authentication. Having in mind this system is targeted as an addon to future final degree dissertations, it shall be an easy to use application which adapts well to any software system.

Therefore, we will be developing a module by means of the Java programming language, with the help of the ESAPI library. This library is free and provides all the security tools that any web developer needs to secure its site [13].

ESAPI is a library developed by OWASP. This allows the generation of a secure web code. The library offers functions that enable validation and attack detection. It is also available for use in other programming languages such as ASP.NET, PHP, ColdFusion, JavaScript, Ruby and Python.

Among some of the enterprises implement ESAPI for the security of their web systems, for example American Express, the International Bank and the US Navy.

I have chosen this tool because of its easy integration with many programming languages. Among the Java libraries that offered similar benefits, we selected ESAPI because it is the most complete and versatile from all the ones examined. [15]

In the case we were dealing with an application with a larger scope, the recommendation would be to use an IAM, since it allows a larger independence and a better management of large user databases. In addition, it allows to monitor the access of users to their accounts. An example of an IAM to be used could be Keycloak. [16]

User Requirements, Software Requirements, traceability matrix and operational environment.

Before finishing with the design part, we must cover the user requirements for the module. These requirements list the functionalities that the software must be deliver to the user. Then, these requirements are used to extract a lower level set of requirements, software requirements, which will allow us to have in mind the mapping between user-level functionalities and how these are to be implemented.

Last, by means of a traceability matrix, we will be checking whether the user requirements have been successfully mapped to a software requirement, and thus, we will somehow reflect every requirement in the final program. Finally, we describe the operational environment.

Design

Next, I will cover the different design decisions taken during the development of the software application. Here you will find some of the main characteristics of the module, such as the architecture, the use cases of the system, sequence diagrams, data design and interface design.

We will describe the architecture:

- Model:
  - Holds the information of the system.
  - Holds the bussiness logic.
  - Extracts the data required by the View.
- View:
  - It is the layer seen by the end user.
  - Asks for information to the model and shapes the returned data in a comprehensive visual way.
- Controller:
  - Interacts with both View and Controller, allowing the correct function of the system as a whole.

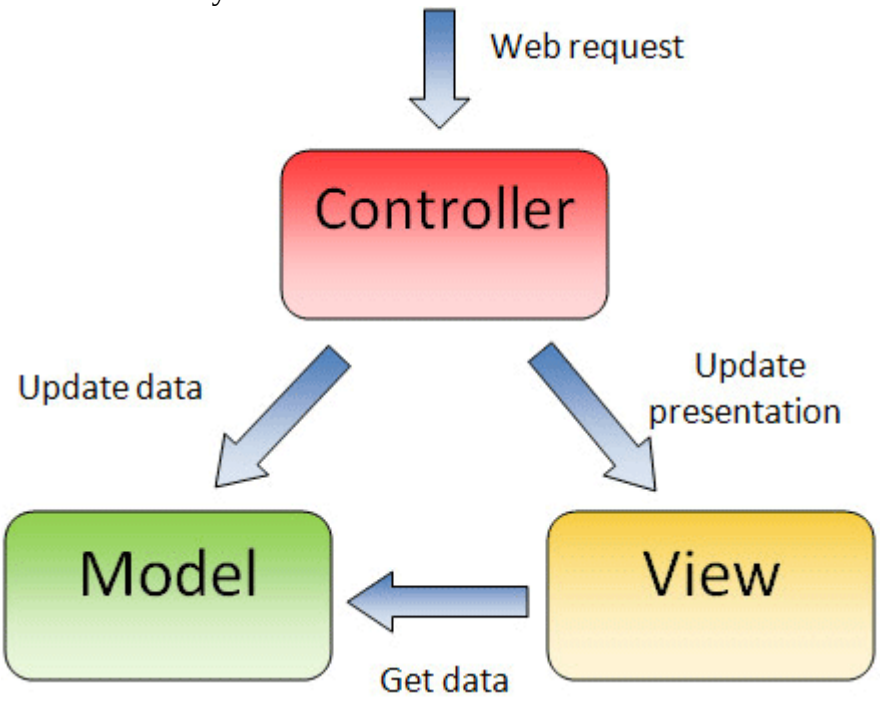

Ilustración 66: Model View Controller architecture scheme. [17]

This architectural scheme was selected since it the system will have an interface that eases the use of it to the end user of the module. The module will also constitute a place to hold the data from the users and will support all user-related functionalities described previously in this paper.

In the system, the view is composed of the JSP pages that will create the UI (User Interface). On the other hand, the model is composed of a file that stores the data and their modifications. Finally, the controller will handle the functions to be performed to appropriately treat user data. So basically, we will have a system consisting of a servlet and auxiliary methods that make sure everything is working properly.

This section will be completed by adding the use cases that will show the system within a real context, performing their functions. In order to improve the comprehension of the use cases, each use case scheme will be accompanied by a sequence diagram. Lastly, I will present the data design and the interfaces offered by the system.

Implementation

To code the solution, we have already mentioned the use of the ESAPI library, which allows to protect the authentication of users in an easy way for most dissertation-alike projects. The final program is based on the Swingset ESAPI tutorial. It is also worth mentioning the fact that our system is comprised by a servlet (controller.java) and a set of JSP pages.

Regarding the controller.java file, it has been created from scratch. This class implements most of the core functionalities of the system, by means of hidden ids that are obtained from the JSPs.

Finally, each JSP has a specific set of operations related to it; for example, there is a JSP which allows to gather user data and send it to the servlet to record its data.

Module evaluation

In order to perform an evaluation of the system build, we will develop a set of use cases, providing us with functional tests, so that we can rapidly check that all the system requirements are fulfilled after the development process. Next, we show a template that is to be used later in the use cases and at the end, we will show a traceability matrix showing that all the compliance with all requirements was successfully checked.

Project management

To guarantee the correct development process of the module, a project plan was made before it was started. This plan was made based upon the dissertation guidelines and hours estimated for the completion of this project. I reserved a 10 days margin for reviews and the plan was finished. The project was estimated to be carried out in 24 weeks, having devoted to it 12 hours and a half per week.

This planning step is very useful to compute the budget of our project, which at the end resulted in 3.271,85€ of total cost, based on the hours devoted and the required material.

If we focus on the economic impact of the project, we can safely state that this is a cheap project, developed upon open source libraries and using Java code, so no additional licenses were required to complete the program.

Regarding the social impact, the data protection is a topic whose awareness should be promoted, to both users of computer systems and developers.

Conclusions

All in all, I reflect in this section about the positive aspects that this work has brought to life, problems encountered and personal comments regarding the process of developing a dissertation like the present one. I will be also providing some light regarding possible future improvements that could be implemented in the security module covered here.

This project has allowed to deepen my expertise within the authentication and protection of credential data, and the diverse ways of doing so. By exploring the state of the art, I have noticed the constant the evolution of the approaches for treating user sensitive data. Moreover, as we have seen, as time goes by, new and improved security

mechanisms join the security world, such as the IAM, paving the ground for developers not familiarized with security concepts.

Regarding the original objectives proposed in the introduction, we can state they have been accomplished. I have been able to keep the system conceptually simple and at the same time, it has been proven to allow the basic functionalities. The interface is very simple so that the user can move around windows in an intuitive way. With this intuition-driven design we aim to promote the reuse of this software in future creations.

The main problem encountered along the development process was to fully understand the functioning of the used library (ESAPI). It is in fact an easy to user library, but as everything, it is harsh at the very beginning to get used to its wonders. Once you get to install the tutorial provided by the creators, you can easily get a handle on the problem.

In conclusion, the outcome of this project is positive. Having to face a complex problem on your own and ending up solving it is what really knowledge is about. For example, in this work we have done some research to get an idea about what kind of approach would fit our final solution, as well as read documentation to get to this point. In addition, having the opportunity to work in a small project like this settles the basics for better facing future challenges.

Future work

Once done with the project, there are some aspects of the module that should be polished, or may be subject to improvements, which were not implemented due to the limited amount of time provided for delivery of this paper.

Now, I enumerate some of this aspects:

- Add functionalities to handle role-based authentication: this feature was not included because it made the whole system more complex.
- Introduce a database: our system does not use a database because the amount of data handled is very poor, so no database was used. In case we were dealing with more users and functionalities such as the formerly describe one, we should be using a database.
- Improve the UI: our system could be tailored to integrate seamlessly in different projects this module integrates with.
- Publish the project repository: The day of the defense, the project that is currently in private mode will be published in the following URL: https://bitbucket.org/Josepaol/tfg-jose-luis/src/master/.

ANEXO V: Bibliografía

[1] INCIBE, «Gestión de credenciales en sistemas de control,» CERTSI, 04 05 2017. [En línea]. Available: https://www.certsi.es/blog/gestion-credenciales-sistemas-control. [Último acceso: 20 04 2018].

[2] M. Rouse, «Identity and access management (IAM),» techtarget, 11 2017. [En línea]. Available: https://searchsecurity.techtarget.com/definition/identity-access-management-IAM-system. [Último acceso: 25 04 2018].

[3] R. Campa Castillo, «UNIDAD 3. AUTENTICACIÓN,» 30 5 2012. [En línea]. Available: http://iscseguridad.blogspot.com/2012/05/unidad-3-autenticacion_30.html. [Último acceso: 26 04 2018].

[4] Wikipedia, «MS-CHAP,» 19 12 2016. [En línea]. Available: https://es.wikipedia.org/wiki/MS-CHAP. [Último acceso: 2018 04 26].

[5] INTEL, «Descripción de la seguridad,» [En línea]. Available: http://support.elmark.com.pl/rgd/drivery/U12C/WLAN/Win7/Docs/ESN/overview.htm. [Último acceso: 10 09 2018].

[6] chakray, «¿Qué es el Single Sign on (SSO)? Definición, características y ventajas,» 30 5 2017. [En línea]. Available: https://www.chakray.com/que-es-el-single-sign-on-sso-definicion-caracteristicas-y-ventajas/. [Último acceso: 1 05 2018].

[7] Wikipedia, «OpenID,» 21 11 2017. [En línea]. Available: https://es.wikipedia.org/wiki/OpenID. [Último acceso: 02 05 2018].

[8] SAML: Qué es, para qué se usa, cómo funciona, «SAML: Qué es, para qué se usa, cómo funciona,» [En línea]. Available: https://cioperu.pe/articulo/24726/saml-que-es-para-que-se-usa-como-funciona/?p=2. [Último acceso: 5 5 2018].

[9] M. Miglani, «Introduction to single sign on (SSO) and SAML,» 8 12 2017. [En línea]. Available: https://kloudrac.com/blog/introduction-to-single-sign-on-sso-and-saml/. [Último acceso: 5 5 2018].

[10] D. Cantón, «Seguridad en OAuth 2.0,» CERTSI, 14 5 2014. [En línea]. Available: https://www.certsi.es/blog/seguridad-oauth-2-0. [Último acceso: 7 5 2018].

[11] A. Llontop, «¿Qué es OAuth y cómo funciona? Esto es lo que necesitas saber,» 18 8 2017. [En línea]. Available: https://adictec.com/que-es-oauth-y-como-funciona/. [Último acceso: 7 5 2018].

[12] J. A. Martin y J. K. Waters, «What is IAM? Identity and access management explained,» csoonline, 17 1 2018. [En línea]. Available:

https://www.csoonline.com/article/2120384/identity-management/what-is-iam-identity-and-access-management-explained.html. [Último acceso: 12 05 2018].

[13] OWASP, «Proyecto OWASP API de seguridad empresarial (ESAPI),» OWASP, 7 4 2010. [En línea]. Available: https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API/es. [Último acceso: 12 5 2018].

[14] P. Dominguez, «En qué consiste el modelo en cascada,» [En línea]. Available: https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada. [Último acceso: 5 7 2018].

[15] M. Maurini, «OWASP ESAPI - Enterprise Security API,» [En línea]. Available: http://tecnologiasweb.blogspot.com/2011/01/owasp-esapi-enterprise-security-api.html. [Último acceso: 10 9 2018].

[16] keycloak, [En línea]. Available: https://www.keycloak.org/.

[17] F. Quintero, «Modelo Vista Controlador (MVC) ejemplo con figuras en C#,» [En línea]. Available: http://fabianquinteropuntonet.blogspot.com/2013/08/modelo-vista-controlador-mvc-ejemplo.html. [Último acceso: 13 8 2018].

[18] ganttproject, «ganttproject,» [En línea]. Available: https://www.ganttproject.biz/.

[19] Gobierno de España, «Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.,» 14 12 1999. [En línea]. Available: https://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750.

[20] R. García Frutos, «Optimización de consultas en bases de datos relacionales,» 2016. [En línea]. Available: https://scholar.google.com/scholar?q=allintitle%3A%22Optimizaci%C3%B3n+de+consultas+en+bases+de+datos+relacionales%22.

[21] A. P. Rivero Ortiz, «Desarrollo de gestor de comidas a domicilio mediante la aplicación de mensajería instantánea Telegram,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+gestor+de+comidas+a+domicilio+mediante+la+aplicación+de+mensajería+instantánea+Telegram%22.

[22] A. Bernárdez Martínez, «Modelado de un puerto marítimo con Unity 3D,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Modelado+de+un+puerto+marítimo+con+Unity+3D%22.

[23] S. Barbero Rodríguez, «Continuous authentication based on data from smart devices,» 2017. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Continuous+authentication+based+on+data+from+smart+devices%22.

[24] J. Valdes Fernández, «Aplicación nutricional y seguimiento deportivo, en atletas de alto rendimiento,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Aplicación+nutricional+y+seguimiento+deportivo%2C+en+atletas+de+alto+rendimiento%22.

[25] L. Lacadena Pascual, «Optimización en Facebook de contenidos especializados en el sector alimentario. Creación de la empresa LULAF.SL,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Optimización+en+Facebook+de+contenidos+especializados+en+el+sector+alimentario.+Creación+de+la+empresa+LULAF.SL%22.

[26] H. Mata San Juan, «Herramienta de análisis de legibilidad de contenidos educativos,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Herramienta+de+análisis+de+legibilidad+de+contenidos+educativos%22.

[27] C. Olivera Fernández-Cortés, «Reducción de dimensionalidad en problemas de regresión,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Reducción+de+dimensionalidad+en+problemas+de+regresión%22.

[28] S. Iriz Ricote, «Técnicas de gestión de vuelo autónomo sobre cuadricóptero,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Técnicas+de+gestión+de+vuelo+autónomo+sobre+cuadricóptero%22.

[29] A. Muñoz Moreno, «Desarrollo de un asistente multimodal educativo para dispositivos móviles Android,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+un+asistente+multimodal+educativo+para+dispositivos+móviles+Android%22.

[30] O. Azzahraoui Daoudi, «Modificación de estados del teléfono usando la tecnología NFC,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Modificación+de+estados+del+teléfono+usando+la+tecnología+NFC%22.

[31] M. Á. Martín García, «Stifman, un agente de concientización en Second Life,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Stifman%2C+un+agente+de+concientización+en+Second+Life%22.

[32] S. Aragonés Tercero, «Estudio de solución basada en sistema NoSQL como sustitución del sistema de directorio usado en el Departamento de Informática,» 2017. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Estudio+de+solución+basada+en+sistema+NoSQL+como+sustitución+del+sistema+de+directorio+usado+en+el+Departamento+de+Informática%22.

[33] H. Santos Senra, «Diseño e implementación de un sistema domótico basado en Raspberry Pi,» 2017. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Diseño+e+implementación+de+un+sistema+domótico++basado+en+Raspberry+Pi%22.

[34] G. Anca Corral, «Clasificación de patrones de siniestros aplicando técnicas de agrupación y segmentación,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Clasificación+de+patrones+de+siniestros+aplicando+técnicas+de+agrupación+y+segmentación%22.

[35] I. Segovia Lasanta, «Estudio de técnicas de reducción de dimensionalidad para problemas supervisados,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Estudio+de+técnicas+de+reducción+de+dimensionalidad+para+problemas+supervisados%22.

[36] E. González González, «Análisis de datos aplicado a siniestros de automóviles,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Análisis+de+datos+aplicado+a+siniestros+de+automóviles%22.

[37] A. Requena López, «Diseño e implementación de una aplicación para la gestión del cortafuegos de Android,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Diseño+e+implementación+de+una+aplicación+para+la+gestión+del+cortafuegos+de+Android%22.

[38] L. Uguina Gadella, «C-mulator. Design and development of an educational web application for teaching C language,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22C-mulator.+Design+and+development+of+an+educational+web+application+for+teaching+C+language%22.

[39] M. San José de Vicente, «Desarrollo de una aplicación de turismo gastronómico para dispositivos iOS y análisis estadístico,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+una+aplicación+de+turismo+gastronómico+para+dispositivos+iOS+y+análisis+estadístico%22.

[40] N. Mertanen Cuní, «Localízame for Android: sistema de localización de dispositivos móviles basado en Android,» 2013. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Localízame+for+Android%3A+sistema+de+localización+de+dispositivos+móviles+basado+en+Android%22.

[41] A. Rossignoli Martínez-Vara del Rey, «Desarrollo de terapias de rehabilitación motora teleoperadas con el robot NAO,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+terapias+de+rehabilitación+motora+teleoperadas+con+el+robot+NAO%22.

[42] H. Lee Marco, «Oracle Business Intelligence for the entreprise,» 2014. [En línea]. Available:

http://scholar.google.com/scholar?q=allintitle%3A%22Oracle+Business+Intelligence+for+the+entreprise%22.

[43] A. Rodríguez Jardón, «Diseño y desarrollo de una aplicación en Android para la evaluación del rendimiento físico,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Diseño+y+desarrollo+de+una+aplicación+en+Android+para+la+evaluación+del+rendimiento+físico%22.

[44] M. Á. Sánchez Valhondo, «Reconocimiento de tipos de hojas, una aplicación de visión artificial de Android,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Reconocimiento+de+tipos+de+hojas%2C+una+aplicación+de+visión+artificial+de+Android%22.

[45] D. Hernández Cassel, «Gestión de servicios IT mediante códigos QR,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Gestión+de+servicios+IT+mediante+códigos+QR%22.

[46] F. Tello Caballo, «Aplicación de HMMs para clasificar series temporales,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Aplicación+de+HMMs+para+clasificar+series+temporales%22.

[47] J. M. Espinosa Montero, «Editor de vídeo de múltiples fuentes para eventos sociales,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Editor+de+vídeo+de+múltiples+fuentes+para+eventos+sociales%22.

[48] P. L. Sanchez Faure, «Razonamiento heurístico para fusión robusta de datos en contexto marítimo,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Razonamiento+heurístico+para+fusión+robusta+de+datos+en+contexto+marítimo%22.

[49] S. Martín Morales, «Análisis de información proveniente de redes sociales como Twitter,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Análisis+de+información+proveniente+de+redes+sociales+como+Twitter%22.

[50] S. Núñez Pulgar, «Un agente en juego Diplomacy,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Un+agente+en+juego+Diplomacy%22.

[51] J. Vázquez Coll, «Simulación de trayectorias de barcos y aplicación al control marítimo,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Simulación+de+trayectorias+de+barcos+y+aplicación+al+control+marítimo%22.

[52] C. Muñoz Villar, «WeSweat, Geolocalización social de deportistas,» 2015. [En línea]. Available:

http://scholar.google.com/scholar?q=allintitle%3A%22WeSweat%2C+Geolocalización+social+de+deportistas%22.

[53] A. Gómez Sanz, «Desarrollo de una aplicación multimodal para la consulta de loterías en dispositivos móviles Android,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+una+aplicación+multimodal+para+la+consulta+de+loterías+en+dispositivos+móviles+Android%22.

[54] A. Arnaiz García, «Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Aplicación+web+para+la+adquisición+colaborativa+de+conocimiento+sobre+Fitopatología+Bacteriana%22.

[55] I. Sáez Lahidalga, «Jugando al 2048 con Inteligencia Artificial,» 2016. [En línea]. Available: https://scholar.google.com/scholar?q=allintitle%3A%22Jugando+al+2048+con+Inteligencia+Artificial%22.

[56] L. Fidalgo Manchón, «Categorización de textos científicos mediante aprendizaje automático,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Categorización+de+textos+científicos+mediante+aprendizaje+automático%22.

[57] J. Urdiales de la Parra, «Desarrollo de aplicación de seguridad vial en Android,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+aplicación+de+seguridad+vial+en+Android%22.

[58] D. Mateos Vázquez, «Técnicas de predicción para energía renovable,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Técnicas+de+predicción+para+energía+renovable%22.

[59] R. Pintos López, «Monitorización del aprendizaje en redes de neuronas,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Monitorización+del+aprendizaje+en+redes+de+neuronas%22.

[60] Á. García-Capelo Blanco, «Modelado y simulación de trayectorias navales y su representación en Unity,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Modelado+y+simulación+de+trayectorias+navales+y+su+representación+en+Unity%22.

[61] J. Corominas Balseyro, «Plataforma web basada en la influencia de la climatología sobre el IBEX 35,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Plataforma+web+basada+en+la+influencia+de+la+climatología+sobre+el+IBEX+35%22.

[62] C. Pantoja Iniesta, «Sistema multiagente para la evacuación de edificios,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Sistema+multiagente+para+la+evacuación+de+edificios%22.

[63] R. Mani Ruiz, «Enfoque de proyecto de implantación de una solución IT para la Gestión del Mantenimiento de Flota en una Empresa Industrial,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Enfoque+de+proyecto+de+implantación+de+una+solución+IT+para+la+Gestión+del+Mantenimiento+de+Flota+en+una+Empresa+Industrial%22.

[64] V. García Cazorla, «Valoración de startups con Aprendizaje Automático,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Valoración+de+startups+con+Aprendizaje+Automático%22.

[65] A. García Herías, «Sistema móvil de información y guiado,» 2013. [En línea]. Available: https://e-archivo.uc3m.es/handle/10016/22589.

[66] M. Pérez Ferro, «Diseño y desarrollo de un cliente y un servidor en JavaScript para gestionar batallas y campeonatos entre agentes inteligentes (JSWARS),» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Diseño+y+desarrollo+de+un+cliente+y+un+servidor+en+JavaScript+para+gestionar+batallas+y+campeonatos+entre+agentes+inteligentes+%28JSWARS%29%22.

[67] A. Pacheco Mayayo, «Desarrollo de una aplicación software para laboratorios remotos : control remoto de prácticas (RLF web),» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+una+aplicación+software+para+laboratorios+remotos+%3A+control+remoto+de+prácticas+%28RLF+web%29%22.

[68] F. A. Castelo Sagnotti, «Plan de empresa de una compañía tecnológica y desarrollo de su primer producto,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Plan+de+empresa+de+una+compañía+tecnológica+y+desarrollo+de+su+primer+producto%22.

[69] P. Sánchez-Herrero Gómez, «Visión artificial integrada con dispositivos de realidad virtual inmersiva aplicada a videojuegos,» 2013. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Visión+artificial+integrada+con+dispositivos+de+realidad+virtual+inmersiva+aplicada+a+videojuegos%22.

[70] C. M. González Escobosa, «Minería de procesos : en ambientes sensorizados,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Minería+de+procesos+%3A+en+ambientes+sensorizados%22.

[71] Á. Montero Casarejos, «Predicción de quiebras empresariales mediante inteligencia artificial,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Predicción+de+quiebras+empresariales+mediante+inteligencia+artificial%22.

[72] P. Fernández Declara, «Cliente Twitter con compresión de datos,» 2013. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Cliente+Twitter+con+compresión+de+datos%22.

[73] A. Escobedo de Pelsmaeker, «Métodos de estimación y análisis de la curva Cupón Cero para el Euro,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Métodos+de+estimación+y+análisis+de+la+curva+Cupón+Cero+para+el+Euro%22.

[74] S. Crespo Toubes, «Aplicación móvil de geolocalización de mercancía bajo los estándares de comercio electrónico militares Foreign Military Sales (FMS) y STANAG 4329,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Aplicación+móvil+de+geolocalización+de+mercancía+bajo+los+estándares+de+comercio+electrónico+militares+Foreign+Military+Sales+%28FMS%29+y+STANAG+4329%22.

[75] A. Antón Aguilar, «Optimización de carteras de inversión mediante técnicas evolutivas y diferentes medidas de riesgo,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Optimización+de+carteras+de+inversión+mediante+técnicas+evolutivas+y+diferentes+medidas+de+riesgo%22.

[76] P. Fernández González, «Herramienta de gestión para elaboración de cuadros de mando,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Herramienta+de+gestión+para+elaboración+de+cuadros+de+mando%22.

[77] D. Muñoz Herrero, «Aplicación móvil para la comunicación interna de una empresa,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Aplicación+móvil+para+la+comunicación+interna+de+una+empresa%22.

[78] J. L. Jiménez Fontenla, «Aplicación móvil para la captura desatendida de datos de sensores en teléfonos inteligentes,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Aplicación+móvil+para+la+captura+desatendida+de+datos+de+sensores+en+teléfonos+inteligentes%22.

[79] P. Carra García, «Predictor en tiempo real de patrones armónicos,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Predictor+en+tiempo+real+de+patrones+armónicos%22.

[80] E. Correas Montiel, «Generación Automática de Editores y Repositorios de Evidencias a partir de Modelos de Estándares de Seguridad,» 2016. [En línea].

Available: http://scholar.google.com/scholar?q=allintitle%3A%22Generación+automática+de+editores+y+repositorios+de+evidencias+a+partir+de+modelos+de+estándares+de+seguridad%22.

[81] S. Morillejo González, «Fraud prevention through segregation of duties: authorization model in SAP GRC Access Control,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Fraud+prevention+through+segregation+of+duties%3A+authorization+model+in+SAP+GRC+Access+Control%22.

[82] S. Pérez Olivares, «Desarrollo de una aplicación esteganográfica para Android,» 2013. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+una+aplicación+esteganográfica+para+Android%22.

[83] A. Peral Rodrigo, «Análisis de situación y propuestas de mejora para el departamento de Service Delivery,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Análisis+de+situación+y+propuestas+de+mejora+para+el+departamento+de+Service+Delivery%22.

[84] V. Martínez Fuertes, «Introducción a la plataforma Arduino y al Sensor ultrasónico HC-SR04,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Introducción+a+la+plataforma+Arduino+y+al+Sensor+ultrasónico+HC-SR04+%3A+experimentado+en+una+aplicación+para+medición+de+distancias%22.

[85] I. Rodríguez López, «Desarrollo de una aplicación de cifrado de imágenes en el sistema Android,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+una+aplicación+de+cifrado+de+imágenes+en+el+sistema+Android%22.

[86] B. Navas Torres, «Desarrollo de una plataforma social para el suministro colaborativo de piezas de repuesto,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+una+plataforma+social+para+el+suministro+colaborativo+de+piezas+de+repuesto%22.

[87] I. Romera Alcalá, «Parkineo, aplicación Android para la búsqueda de parking,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Parkineo%2C+aplicación+Android+para+la+búsqueda+de+parking%22.

[88] K. El Maataoui, «Diseño, desarrollo e implantación de una plataforma empotrada para el control de sistemas robóticos,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Diseño%2C+desarrollo+e+implantación+de+una+plataforma+empotrada+para+el+control+de+sistemas+robóticos%22.

[89] F. Caro Herranz, «Sistemas de construcción de mapas en PDDL para la planificación automática,» 2013. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Sistemas+de+construcción+de+mapas+en+PDDL+para+la+planificación+automática%22.

[90] C. E. Pérez Moscoso, «Estrategias de diversificación eficiente de carteras e implementación de una plataforma digital de inversión,» 2015. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Estrategias+de+diversificación+eficiente+de+carteras+e+implementación+de+una+plataforma+digital+de+inversión%22.

[91] C. Manzano Carrasco, «Interacción humano-robot con el robot REEM sobre el framework RoboComp,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Interacción+humano-robot+con+el+robot+REEM+sobre+el+framework+RoboComp%22.

[92] S. Torres Mendiola, «Desarrollo de una aplicación para la gestión de proyectos no gubernamentales,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Desarrollo+de+una+aplicación+para+la+gestión+de+proyectos+no+gubernamentales%22.

[93] E. Sánchez Checa, «Plan de negocio de empresa basada en Internet de las Cosas y el lanzamiento de un producto: regulador de puerta de garaje por reconocimiento de matrículas de coche mediante Raspberry Pi,» 2014. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Plan+de+negocio+de+empresa+basada+en+Internet+de+las+Cosas+y+el+lanzamiento+de+un+producto%3A+regulador+de+puerta+de+garaje+por+reconocimiento+de+matrículas+de+coche+mediante+Raspberry+Pi%22.

[94] O. Cabezas Velasco, «SiGUP, sistema de gestión de usuarios para una plataforma distribuída de control de proyectos software,» 2013. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22SiGUP%2C+sistema+de+gestión+de+usuarios+para+una+plataforma+distribuída+de+control+de+proyectos+software%22.

[95] M. d. C. d. Barrio Cerro, «Técnicas de computación evolutiva aplicadas a la clasificación a partir de monitores de actividad física,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Técnicas+de+computación+evolutiva+aplicadas+a+la+clasificación+a+partir+de+monitores+de+actividad+física%22.

[96] D. Suárez Esteban, «Diseño y desarrollo de una herramienta software para la creación de contenidos de realidad aumentada orientada a usuarios finales (end users),» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Diseño+y+desarrollo+de+una+herramienta+software+para+la+creación+de+contenidos+de+realidad+aumentada+orientada+a+usuarios+finales+%28end+users%29%22.

[97] V. González Sánchez, «Patrones de paralelismo: una aproximación basada en bibiliotecas génericas,» [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Patrones+de+paralelismo%3A+una+aproximación+basada+en+bibiliotecas+génericas%22.

[98] V. M. Zamora España, «Gestión de rutas y toma de decisiones en el entorno de simulación STI-SIM,» 2016. [En línea]. Available: http://scholar.google.com/scholar?q=allintitle%3A%22Gestión+de+rutas+y+toma+de+decisiones+en+el+entorno+de+simulación+STI-SIM%22.